

No. 16-2140

**In the
United States Court of Appeals
for the Federal Circuit**

IN RE: DROPLETS, INC.,
Appellant.

Appeal from the United States Patent and Trademark Office,
Patent Trial and Appeal Board, Reexamination No. 95/002,330

OPENING BRIEF FOR APPELLANT DROPLETS, INC.

Timothy J. Bechen
BECHEN PLLC
9200 Forest Hill Ave., Suite 4-C
Richmond, VA 23235
Tel.: (804) 928-7037
tim@bechenlaw.com

Daniel L. Geyser
Principal Attorney
STRIS & MAHER LLP
725 S. Figueroa Street, Suite 1830
Los Angeles, CA 90017
Tel.: (213) 995-6811
Fax: (213) 261-0299
daniel.geyser@strismaher.com

Counsel for Appellant

No. 16-2140

**In the
United States Court of Appeals
for the Federal Circuit**

IN RE: DROPLETS, INC.,
Appellant.

Appeal from the United States Patent and Trademark Office,
Patent Trial and Appeal Board, Reexamination No. 95/002,330

CERTIFICATE OF INTEREST

Pursuant to Fed. Cir. R. 47.4, counsel for Appellant Droplets, Inc. certifies the following:

1. The full name of every party or amicus represented by me is:

Droplets, Inc.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

None.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party represented by me are:

None.

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this Court are:

Bechen PLLC: Timothy J. Bechen.

Stris & Maher LLP: Daniel L. Geyser.

Respectfully submitted.

/s/ Daniel L. Geyser

Daniel L. Geyser

Principal Attorney

STRIS & MAHER LLP

725 S. Figueroa Street, Suite 1830

Los Angeles, CA 90017

Tel.: (213) 995-6811

Fax: (213) 261-0299

daniel.geyser@strismaher.com

Counsel for Appellant Droplets, Inc.

December 9, 2016

TABLE OF CONTENTS

	<u>Page</u>
JURISDICTIONAL STATEMENT	1
STATEMENT OF THE ISSUES.....	1
STATEMENT OF THE CASE.....	2
SUMMARY OF ARGUMENT	6
STANDARD OF REVIEW	7
ARGUMENT	8
I. THE BOARD ERRED IN FINDING CLAIMS 6 AND 20 INVALID AS ANTICIPATED OR OBVIOUS	8
A. Frese And Hickman Fail To Teach Multiple Aspects Of The Claimed Invention	8
1. Neither reference teaches “re-establishing” the communication connection to the host computer.....	9
2. Hickman does not disclose “storing” the required link	11
3. Hickman’s so-called “link” is not even a link, but a rudimentary URL.....	15
B. Frese And Windows Secrets Fail To Teach Multiple Aspects Of The Claimed Invention	16
C. ArcView Fails To Teach Multiple Aspects Of The Claimed Invention.....	20
D. ArcView And Orenshteyn Fail To Teach Multiple Aspects Of The Claimed Invention—And Orenshteyn Is Not Even Prior Art	25

TABLE OF CONTENTS
(continued)

	<u>Page</u>
II. THE BOARD’S INTER PARTES REEXAMINATION IS UNCONSTITUTIONAL UNDER ARTICLE III AND THE SEVENTH AMENDMENT	28
CONCLUSION	29

TABLE OF AUTHORITIES

	<u>Page(s)</u>
 <u>Cases</u>	
<i>Advanced Display Sys., Inc. v. Kent State Univ.</i> , 212 F.3d 1272 (Fed. Cir. 2000).....	20
<i>Arendi S.A.R.L. v. Apple Inc.</i> , 832 F.3d 1355 (Fed. Cir. 2016).....	7, 14
<i>Cutsforth, Inc. v. MotivePower, Inc.</i> , 636 F. App'x 575 (Fed. Cir. 2016)	6, 26
<i>Granfinanciera, S.A. v. Nordberg</i> , 492 U.S. 33 (1989).....	28
<i>In re Baker Hughes, Inc.</i> , 215 F.3d 1297 (Fed. Cir. 2000).....	14
<i>In re Gartside</i> , 203 F.3d 1305 (Fed. Cir. 2000).....	8
<i>In re Giannelli</i> , 739 F.3d 1375 (Fed. Cir. 2014).....	7
<i>In re Gleave</i> , 560 F.3d 1331 (Fed. Cir. 2009)	8
<i>In re Lee</i> , 277 F.3d 1338 (Fed. Cir. 2002)	7, 14, 28
<i>In re Mouttet</i> , 686 F.3d 1322 (Fed. Cir. 2012)	7
<i>In re Zurko</i> , 258 F.3d 1379 (Fed. Cir. 2001)	14
<i>Joy Techs. v. Manbeck</i> , 959 F.2d 226 (Fed. Cir. 1992).....	29
<i>MCM Portfolio LLC v. Hewlett-Packard Co.</i> , 812 F.3d 1284 (Fed. Cir. 2015).....	29
<i>Microsoft Corp. v. Proxyconn, Inc.</i> , 789 F.3d 1292 (Fed. Cir. 2015)	13
<i>Motor Vehicle Mfrs. Ass'n v. State Farm Mut. Auto. Ins. Co.</i> , 463 U.S. 29 (1983)	20
<i>Nike, Inc. v. Adidas AG</i> , 812 F.3d 1326 (Fed. Cir. 2016)	19
<i>Power Integrations, Inc. v. Lee</i> , 797 F.3d 1318 (Fed. Cir. 2015)	7
<i>Regents of Univ. of Minn. v. AGA Med. Corp.</i> , 717 F.3d 929 (Fed. Cir. 2013).....	13
<i>Spansion, Inc. v. ITC</i> , 629 F.3d 1331 (Fed. Cir. 2010)	21

TABLE OF AUTHORITIES
(continued)

	<u>Page(s)</u>
<i>Stern v. Marshall</i> , 564 U.S. 462 (2011).....	28
<i>Synopsys, Inc. v. Mentor Graphics Corp.</i> , 814 F.3d 1309 (Fed. Cir. 2016)	20

Statutes

28 U.S.C. 1295(a)(4)(A)	1
35 U.S.C. 102(b)	2, 20
35 U.S.C. 103(a)	2, 8, 25
35 U.S.C. 134(b)	1
35 U.S.C. 141(c)	1
35 U.S.C. 142	1
35 U.S.C. 311	1
35 U.S.C. 316(c)	2

Regulations

37 C.F.R. 90.3(a)(1)	1
----------------------------	---

STATEMENT OF RELATED CASES

Pursuant to Fed. Cir. R. 47.5, Appellant Droplets, Inc. respectfully submits that:

(a) There have been no other previous appeals in this case.

(b) There are two related cases (or sets of cases) that may directly affect or be directly affected by this Court's decision in the pending appeal. First, *Droplets, Inc. v. E*TRADE Bank, et al.*, Nos. 16-2504 & 16-2602 (Fed. Cir.), involves a related cross-appeal to another PTAB proceeding challenging the validity of a patent in the same family as the '838 patent at issue here. Second, there are related cases pending in the following district courts (all of which have been stayed pending the disposition of this Court's review): (i) *Droplets, Inc. v. E*TRADE Fin. Corp., et al.*, No. CV 1:12-cv-02326-CM (S.D.N.Y); (ii) *Droplets, Inc. v. Nordstrom, Inc.*, No. 3:12-cv-04049-JST (N.D. Cal.); and (iii) *Droplets, Inc. v. Amazon, Inc., et al.*, No. 3:12-cv-03733-JST (N.D. Cal.).

JURISDICTIONAL STATEMENT

The U.S. Patent Trial and Appeal Board (Board) had jurisdiction under 35 U.S.C. 134(b) and 35 U.S.C. 315, and this Court has jurisdiction under 28 U.S.C. 1295(a)(4)(A) and 35 U.S.C. 141. The notice of appeal was filed on May 26, 2016, after the Board's final decision was entered on March 30, 2016. Under 35 U.S.C. 142, Droplets' appeal is timely.¹

STATEMENT OF THE ISSUES

In this inter partes reexamination, the Board invalidated all 38 claims of a groundbreaking patent. That patent has fundamentally changed the way the Internet functions. It has achieved commercial success and withstood challenges by industry titans. Out of 20 defendants, 13 settled before trial, and 2 went to trial, lost, and settled later. But in the decision below, the Board struck all 38 claims using 17 different grounds, ruling against Droplets on virtually every point. Its short ruling addressed most of these complex issues in a single sentence—and (on multiple occasions) failed to address Droplets' arguments at all.

In an effort to simplify the appeal, Droplets seeks to revive only two claims—claim 6 and claim 20.

¹ In this brief, Droplets refers to the statutory and regulatory provisions governing inter partes reexamination that were in effect before the America Invents Act (AIA), Pub. L. No. 112-29, 125 Stat. 284 (2011).

The questions presented are:

1. Whether the Board erred in holding that the challenged claims were anticipated or obvious under 35 U.S.C. 102(b) or 103(a).
2. Whether the Board's inter partes reexamination violates Article III and the Seventh Amendment of the U.S. Constitution by allowing Article I judges to adjudicate the validity of an issued patent.

STATEMENT OF THE CASE

1. This case involves a groundbreaking patent that has shaped the modern online world. In 1999, the Internet ran at a fraction of today's speed. Most websites produced static images and offered minimal interactivity. Refreshing online content typically required redrawing and resending an entire page—which was especially burdensome over slow connections with limited bandwidth.

Droplets' inventors sought to change this. They discovered systems and methods for opening the Internet to online applications and interactive communication connections. They divided functionality between clients and servers. Users could now establish communication connections with host computers, accessing web-based applications that would operate like desktop programs. They even discovered ways to share links across multiple devices, re-establishing connections on a variety of platforms and restoring prior sessions to pick up seamlessly where users left off.

Three claims are illustrative:

1. A method for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the method comprising:

retrieving, in response to a request of a client computer, a content item having computer program code embedded therein, execution of the embedded computer program code establishing a communication connection to a host computer;

sending operating environment information regarding the client computer from the client computer to the host computer;

retrieving presentation information to present an application and content, the presentation information being based on the operating environment information and comprising at least one of instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the presented application; and

presenting, at the client computer, the application and the content based upon the presentation information.

2. The method of claim 1 comprising:

storing, on the client computer, a link for re-establishing the communication connection to the host computer; retrieving the presentation information; and presenting the application and the content.

6. The method of claim 2 wherein storing further comprises storing instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.

Appx693 (U.S. Patent No. 7,502,838 (29:48-30:2; 30:3-7; 30:18-23)).²

² This case also involves claim 20, which is an apparatus claim that closely tracks method claim 6. Appx694 (31:39-45).

Droplets operates as a software-development company. Its patented technology has been licensed by industry giants and its customer list has included both major governmental entities and private operations. Its patents have been tested in litigation and survived invalidity challenges in high-stakes situations. Its patented technology has now also been challenged before the USPTO.

2. This matter began below when Google and Facebook petitioned for inter partes reexamination. Appx697. The initial petition was massive and sprawling. Petitioners sought reexamination of all 38 claims in a 111-page filing. Appx702-812. The petition included 48 claim charts covering 1286 pages. Appx3454-4739. It argued 46 separate grounds of rejections with multiple independent references and combinations of alleged prior art. The Examiner's original action was hundreds of pages long (between the decision and its attachments). Appx372-631.

But something happened near the beginning: Google and Facebook almost immediately dropped out. This left a difficult dynamic: due to petitioners' early withdrawal, this matter became effectively an ex parte reexamination under the strict inter partes rules. Rather than waiting for petitioners to narrow their submission, Droplets was forced to engage everything, but without the benefit of ordinary Office interaction.

The Examiner eventually rejected all 38 claims in the '838 patent (Appx27), and Droplets appealed to the Board.

3. The Board affirmed. Appx1-26. Its opinion swept through 17 grounds of rejections (most involving multiple references) in a short 24 pages. It dispatched many issues in a single sentence, and released drive-by claim constructions—even where the Board’s construction was at odds with competing rulings by district courts.

While the decision eliminated the entirety of Droplets’ patent, this appeal will focus only on two critical claims: claims 6 and 20. The Board declared those claims obvious or anticipated on four grounds, the vast majority of which overlapped. These are the five references at issue:

- Frese (U.S. Patent No. 5,909,545) discloses a software-demonstration system to let users try software programs over the Internet before purchase. The system uses mirroring applications on the client and the server to “trick” the server application into thinking it is being executed on a client device. Appx843-860.
- Hickman (U.S. Patent No. 6,173,332) is a system for selecting one of a plurality of network computers to communicate with a single client computer. Appx884-920.
- Windows Secrets is a published self-help book for using Windows 98. Appx1781.
- The ArcView User Guide discusses a system for delivering map images to a web browser. It involves unitary data calls: a client using a web-browser application requests specific map images; those requests are passed to an intermediate server, which then transmits the requests to a map application on a server; that application then generates the static map image and sends it back to the intermediate server to send the image back to the web browser. Appx957-1020.

- Orenshteyn (U.S. Patent No. 6,393,569) is a continuation-in-part of U.S. Patent No. 5,889,942. It describes a secured system for accessing application services from a remote station. While Orenshteyn’s parent patent was issued before the ’838 priority date, the continuation-in-part was issued later—and all sides agree the new matter is not entitled to priority.

This appeal followed.

SUMMARY OF ARGUMENT

1. The Board demonstrably erred in striking down the entirety of the claims in Droplets’ ’838 patent. Notwithstanding the complexity below, the Board’s missteps are simple and obvious. Each ground is infected with pure errors of law and logic: the Board misconstrued the patent’s critical terms, misread the clear evidence, said prior art disclosed concepts it plainly could not, and provided virtually no explanation to justify its decision—and, at times, provided no explanation at all. Compare *Cutsforth, Inc. v. MotivePower, Inc.*, 636 F. App’x 575, 578 (Fed. Cir. 2016) (“The majority of the [Board]’s Final Written Decision is spent summarizing the parties’ arguments and offers only conclusory analysis of its own. While the decision does specify when it is rejecting a party’s argument, the [Board] does not explain why it accepts the remaining arguments as its own analysis. This leaves little explanation for why the [Board] found the claimed invention obvious.”).

This is the rare appeal from a PTAB ruling where each of the Board’s rejections is manifestly deficient on multiple grounds. The decision below incorrectly struck down a vital patent, and the judgment should be reversed.

2. This IPR proceeding is invalid for a separate reason: it violates Article III and the Seventh Amendment. Those constitutional safeguards preclude forcing Droplets to adjudicate its vested property right (an issued patent) before a non-Article III tribunal. Congress cannot delegate the disposition of private disputes outside the judicial branch. While this Court's decisions have rejected this argument, Droplets respectfully preserves the issue for further review.

STANDARD OF REVIEW

The Board's decision is reviewed under the strictures of the Administrative Procedure Act. The Court reviews the Board's legal conclusions de novo and its factual findings for substantial evidence. *E.g., In re Giannelli*, 739 F.3d 1375, 1378-1379 (Fed. Cir. 2014). "Substantial evidence is something less than the weight of the evidence but more than a mere scintilla of evidence." *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1360 (Fed. Cir. 2016) (quoting *In re Mouttet*, 686 F.3d 1322, 1331 (Fed. Cir. 2012)). "[T]he board is obligated not only to come to a sound decision, but to fully and particularly set out the bases upon which it reached that decision." *Power Integrations, Inc. v. Lee*, 797 F.3d 1318, 1323 (Fed. Cir. 2015).

The ultimate determination of obviousness is a question of law, but that determination is based on underlying factual findings. See, *e.g., In re Gartside*, 203

F.3d 1305, 1316 (Fed. Cir. 2000). Anticipation is a fact question reviewed for substantial evidence. *In re Gleave*, 560 F.3d 1331, 1334 (Fed. Cir. 2009).

When the Board's findings lack substantial evidence, the Court may reverse those findings without remanding for further proceedings. *Giannelli*, 739 F.3d at 1380.

ARGUMENT

I. THE BOARD ERRED IN FINDING CLAIMS 6 AND 20 INVALID AS ANTICIPATED OR OBVIOUS

The Board declared claims 6 and 20 invalid on four grounds, but each is insubstantial. As explained below, the Board repeatedly misconstrued (or misunderstood) the claims, misread the factual record, committed errors of law and logic, and rested on conclusory reasoning—or no reasoning at all. These important claims disclose groundbreaking advances that are not found anywhere in the prior art. The judgment below should be reversed.

A. Frese And Hickman Fail To Teach Multiple Aspects Of The Claimed Invention

Contrary to the Board's determination, claims 6 and 20 are not obvious under 35 U.S.C. 103(a) in light of Frese and Hickman. For multiple reasons, those references fail to disclose critical aspects of the claimed invention.

1. Neither reference teaches “re-establishing” the communication connection to the host computer

a. As a key feature of the invention, the challenged claims require a “link for re-establishing the communication connection to the host computer.” Appx693 (30:4-5). This language is unequivocal: It focuses on restoring the earlier connection; that is why it says “*the* connection” (rather than “*a* connection”) and “*the* host computer” (rather than “*a* host”). Those antecedent terms trace back to the original connection in claim 1—and once that connection is established with the host, claim 2 provides a way to *reestablish* that connection. Appx9000-9001; Appx681 (5:4-9).

The claimed method is accordingly very different from permitting multiple, static, unitary calls to a network or server. The invention provides a means to revive the “communication connection” that was established before—without having to retrace every step back to an original website, and without having to establish a new connection with a different host computer. Appx687 (17:39-44). It does not mean simply stumbling across the same website or the same host computer twice.

b. Neither reference (alone or combined) discloses these concepts.

First, although the Board insists that Hickman discloses these elements, even the Board concedes that Frese does not teach these claims. Frese, again, provides a mirroring function: it permits users to demo an application operated remotely by viewing the application through the client computer. But Frese is a *stateless* sys-

tem. Appx3787-3788. That means each request is an *independent* transaction that is unrelated to any previous request. The server accordingly does not retain information to restore prior connections to the client computer. Each demo session (by design) is independent from the next. This is the very opposite of claim 2: its method requires *re-establishing* the same connection to the host computer, thus resuming the last session (and restoring the state information that Frese, by contrast, does not retain).

Second, contrary to the Board's contention, Hickman does not close the gaps left by Frese. Hickman involves a series of independent calls to different network computers. It does not disclose any means for reestablishing the last connection to the same host—it simply provides a means for establishing new connections to other hosts, which can later upload information *stored remotely* (i.e., not on the client computer) to replicate a past session. Hickman does not teach (and had no *reason* to teach) how to reconnect with a host computer among a plurality of host computers, which is what the '838 patent requires (Appx693 (29:48-51)).

The Board's response was inadequate. It recognized that Hickman failed to disclose any deliberate way to reach the same host to resume a prior session. So the Board contrived a work-around: It reasoned that Hickman contemplates a scenario where the network *might* consist of a single host computer, and in that unusual instance, the only possible subsequent connection is back to that same computer.

Appx10 (the Board’s full reasoning: “the express teaching of Hickman contemplates a cluster with a single network-accessible computer (NAC) rendering each subsequent connection a ‘re-establishing’ of that connection”).

The Board’s logic misses the point entirely. Even in that rare embodiment, Hickman is not technology for *identifying* the host (among a plurality) and reestablishing the past connection; it is simply a way to stumble across the same computer by happenstance. The ’838 patent requires reestablishing the connection by design. It is clearly insufficient to say that *if* a network happens to have a single computer, a connection will automatically revert to that single computer. That hardly discloses the inventive means of tracing a path to the prior computer where a network consists of a “plurality of host computers.”

In any event, the Board overlooked an important point: in that embodiment, Hickman would not satisfy a clear element of claim 1: that the network consists of a “plurality” of host computers, not a single host. So even if Hickman disclosed claim 2’s elements by accident (which it did not), it did not disclose how to accomplish that result in the required conditions of the patented invention.

2. Hickman does not disclose “storing” the required link

The Board was also wrong for another reason: Hickman fails to disclose “storing” the claimed link. The patent’s language is unambiguous: the “link for reestablishing the communication connection” must be “*stor[ed] on the client com-*

puter.” Appx693 (30:4-5) (emphasis added). That express element means it is not enough to create or transmit a “link”—the claim specifically requires the link to be *stored* on the local client.

As above, there is no genuine dispute that Hickman does not “store” links in any traditional sense. So the Board again looked for a work-around: In Hickman, the Board says, a URL of a network computer is provided upon request, and a Java Applet “convert[s]” that URL to display a “link” in the client’s browser. Appx9; see also Appx888, 911-912 (Fig. 3A; 10:45-11:5). According to the Board, the displayed “link” is automatically “stored” merely by “informing” the client computer of the URL. Appx9. This is the entirety of the Board’s reasoning: “we find ‘informing’ a computer of any piece of data necessarily requires ‘storing’ that data within the computer because storing would be necessary for the computer to receive the information.” Appx10. The Board is wrong.

First, the Board plainly misconstrued the claim term. The entire point of *storing* the link was not simply to say that it was lodged in the computer’s transient memory. It was to make the link available “*for re-establishing the communication connection.*” Appx687 (17:14-16) (“the present invention provides a mechanism for locally storing links to remote droplet-enabled applications and/or information”). The link is stored so it later can be accessed and used—not so it invisibly consumes volatile memory behind the scenes.

Moreover, the claim’s plain language is reaffirmed by the specification. The ’838 patent draws a specific distinction between “download[ing]” applications and “stor[ing]” applications. Appx687 (17:20-21). And it does so in a passage explicitly describing the “present invention.” See *Regents of Univ. of Minn. v. AGA Med. Corp.*, 717 F.3d 929, 936 (Fed. Cir. 2013) (“When a patent thus describes the features of the ‘present invention’ as a whole, this description limits the scope of the invention.”) (internal quotation marks omitted). This is directly at odds with the Board’s sweeping presumption that “storing” means virtually anything—and captures “any piece of data” (Appx10) that happens to cross the computer’s path. See, e.g., Appx6582 (the term “storing” requires more than “merely download[ing] or transmit[ing]” data from a server”) (claim-construction order); see also *Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1298 (Fed. Cir. 2015) (“[e]ven under the broadest reasonable interpretation, the Board’s construction cannot be divorced from the specification and the record evidence, and must be consistent with the one that those skilled in the art would reach”).

In any event, the Board’s reading simply *presumes* facts about Hickman and where its “data” is stored. But its presumption is just that—a *presumption*. The Board identified no factual support or anything in Hickman explaining where or how this “link” is stored on the client computer—much less how any such “storage” could make the link available for its intended use (read: making the link

available later for “*re-establishing the communication connection*”). See *In re Lee*, 277 F.3d 1338, 1344 (Fed. Cir. 2002) (“[c]onclusory statements * * * do not fulfill the agency’s obligation”); *In re Zurko*, 258 F.3d 1379, 1386 (Fed. Cir. 2001) (Board “must point to some concrete evidence in the record in support of [its] findings”).

At bottom, the Hickman “link” is entirely transitory; it is not “stored” on the client computer, but temporarily presented in the active web browser. See Appx3788. The Board simply shoehorned Hickman into a foreign context, and then failed to provide any support or detail (evidentiary or otherwise) for the bald assertion that merely *informing* the client computer of the link automatically resulted in that link being *stored* on the client computer. See *Arendi*, 832 F.3d at 1362 (“references to ‘common sense’ * * * cannot be used as a wholesale substitute for reasoned analysis and evidentiary support”).

Under the ’838 patent, however, links are “stored” to permit client computers to re-establish connections with host computers—without first navigating back to a source page to request a URL. The Board’s construction misreads the claim, frustrates its purpose, and presumes key facts without factual support. Hickman does not disclose this critical element of the claimed invention, which is dispositive on this ground. See *In re Baker Hughes, Inc.*, 215 F.3d 1297, 1301 (Fed. Cir. 2000) (PTO’s claim construction reviewed de novo).

3. Hickman’s so-called “link” is not even a link, but a rudimentary URL

The Board’s determination fails for yet another reason: It wrongly characterizes Hickman’s URL as a “link” for purposes of the claimed invention. Appx10 (“We find the described ‘Click Me to Connect’ disclosure within Hickman, in response to the provision of a URL, to be suggestive of the provision of a link.”). The central point of the link—which the Board ignores—is storing processing functionality on the client computer to reinitiate the *same* connection via client-directed activity. Hickman, by contrast, requires navigating to the network, receiving a URL, having a Java Applet convert that URL, and having the user “click” on the target—which is presented, apparently, only while the web browser is open. That understanding of “link” lacks any support in the patent. It is effectively a hyperlink—the kind of prior art the specification distinguished as *not* the invention. See Appx679 (2:52-57); Appx6578 (claim-construction order).

Moreover, the Board has not explained why Hickman’s so-called “link” is imbued with the necessary functionality. Indeed, the Board can only say it is “*suggestive* of the provision of a link.” Appx10 (emphasis added). On a fair reading, the key embodiment in Hickman reflects visual text on a display screen (“Click Me to Connect”); it is not underlying software executable code, and it does not serve the functions of the claimed “links” under the ’838 patent.

For any or all of these reasons, the Board erred in finding the challenged claims invalid in light of Frese and Hickman.

B. Frese And Windows Secrets Fail To Teach Multiple Aspects Of The Claimed Invention

As above, the Board erred in declaring the challenged claims obvious in light of Frese and Windows Secrets. Frese fails for the same reasons discussed above, and Windows Secrets discloses a simple bookmark—which is not a “link” and assuredly does not teach how to *re-establish* a *communication connection* with a particular host computer.

1. As explained above, a simple URL is not a “link” within the meaning of the challenged claims. Yet that is the full scope of Windows Secrets’ disclosure. It explains how to create a fresh, unique HTML call to a static web server; it does not teach how to re-reestablish a communication connection to a specific host computer. Appx3846; Appx1885, 2057.

Put simply: Calling the same website twice (on independent occasions with no connection between each instance) is not *re-establishing* a link. It is merely accessing the same website twice. That functionality was already known in the art (*e.g.*, bookmarks), and the ’838 patent—with its very different structure and approach—was designed to overcome those defects. It is unclear what the Board found “obvious” about novel elements that appear nowhere in either reference.

In response, the Board repeats its position that there is “no patentable distinction between the claimed ‘link’ and the disclosed ‘URL.’” Appx11. But the central point of the patent was *the distinction between the claimed “link” and the prior-art URLs*. That prior art was exactly what the patent was explicitly designed to overcome. Yet the Board and the Examiner rely upon passages in the background section of the specification describing the prior art—while ignoring the later passages explaining the limitations of that art and showing how Droplets’ “links” overcome those limitations. Appx6195; Appx682 (7:39-44). The Board never says what it makes of the *material improvements* to URLs in the claimed invention, if the invention were simply a URL.

Further, we know that static calls to websites are insufficient because the specification itself makes that clear. Again, in its background section, the specification describes static “hyperlinks” that navigate to different websites. But those links did not *establish new connections* (like the patented claims) or permit remote applications to present themselves to the client computer. This invention is about generating a true communication connection so applications on remote servers can effectively operate on a client computer. That central aim cannot be achieved by a simple bookmark to a static website.

The Board finally tried to brush everything aside because Windows Secrets itself apparently describes a “‘link’ as a ‘URL shortcut.’” Appx11. The Board was

confused: The question is how *the '838 patent* defines “link,” not how Windows Secrets defines the term. An “express description” might matter if it came from the patent’s specification. But it makes no difference coming from an outside source.

2. Even if Windows Secrets somehow did teach a “link,” neither reference discloses the critical concept of a link “for re-establishing the communication connection to the host computer.”

First, as explained earlier, Frese self-identifies as a “stateless” application: it uses no client-state information and cannot re-establish past connections—a task unnecessary to Frese’s functionality as a demo program. This is likely why the Board did not even attempt to invoke Frese as suggesting these elements in its earlier rejection.

Here, however, the Board tries a new tack: it says that nothing in Frese “*prohibits* the re-establishing of a communication connection.” Appx11 (emphasis added). But that asks the wrong question: it is not Droplets’ job to show the reference lacks the claimed elements, but the *Board’s* job to show that each claimed element is present. There is nothing in Frese that shows how to store a link to re-establish the same communication connection with the same host computer (much

less restoring the session to its previous state).³ The Board simply shifted the burden of proof to Droplets, which was error. See, *e.g.*, *Nike, Inc. v. Adidas AG*, 812 F.3d 1326, 1334 (Fed. Cir. 2016). Without an *affirmative* showing, the Board’s conclusion lacks substantial evidence.

Second, again, Windows Secrets effectively discloses saving a bookmark. This does not *re-establish* a communication connection; it simply results in re-accessing the same static website twice, each time with a new connection. That does not teach or explain how to reach a host computer in a network of a plurality of host computers, and it does not show how to use client information to restore the connection to its previous state. A static HTML call—even back to the same website—simply does not open a *communication connection* with that site.

In sum, the Board confuses a “link” with a “URL”—the former has functionality that the latter does not. Windows Secrets teaches how to store a URL (without any ability to isolate a former host computer, re-establish a past connection, restore the past operating state of the connection, or, indeed, to open a *com-*

³ In order to re-establish a communication connection, the system needs state information to restore the previous session. See Appx680 (4:1-4) (claim 2 contemplates “links” that “include facilities for restoring previous operating states of the application as the application is re-represented at a user’s computer”). Neither Frese nor Windows Secrets provides any hint about how accessing a static website with a fresh URL call could possibly support “re-establishing” the connection under the claims. And Frese, as a “stateless system,” obviously cannot restore connections to their previous state—indeed, it specifically *disclaims* that ability.

munication connection at all). Frese fails to show any means of re-establishing anything—which Frese does by design *as a stateless system*. The Board wholly failed to explain how either source (together or otherwise) taught each element of the patented claims.

C. ArcView Fails To Teach Multiple Aspects Of The Claimed Invention

According to the Board, the ArcView user guide anticipates the challenged claims under 35 U.S.C. 102(b). See Appx20-21. This rejection fails for multiple reasons, including that ArcView does not disclose each claimed element—*as the Board admits*. See *Advanced Display Sys., Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000) (anticipation requires disclosing each claim element).

1. As Droplets explained, ArcView cannot anticipate without showing “a link for re-establishing the communication connection to the host computer.” The Board acknowledged Droplets’ argument, but failed to address it. Appx20-21. That alone is error: under settled APA standards, the Board must provide a reasoned explanation of its disposition, and silence is the opposite of a reasoned explanation. See, *e.g.*, *Synopsys, Inc. v. Mentor Graphics Corp.*, 814 F.3d 1309, 1322 (Fed. Cir. 2016) (the Board “must articulate logical and rational reasons for [its] decisions”); see also *Motor Vehicle Mfrs. Ass’n v. State Farm Mut. Auto. Ins. Co.*, 463 U.S. 29, 43 (1983) (agencies “must examine the relevant data and articulate a satisfactory

explanation for [their] action including a rational connection between the facts found and the choice made”).

But the Board was not silent for long. In the next section of its opinion, the Board explicitly conceded that ArcView lacks this element: it confessed that “the remote storage of a link for reconnecting to a host computer” is “a feature absent from ArcView.” Appx22. Because that feature is an essential element of the challenged claims, ArcView by definition does not anticipate, and the Board’s contrary finding was baseless. See, *e.g.*, *Spansion, Inc. v. ITC*, 629 F.3d 1331, 1356 (Fed. Cir. 2010).

Moreover, the Board’s concession was correct: ArcView, like Frese, “requires stateless client/server communications.” Appx995. By design, it requires a series of independent calls to a static server. *Ibid.* There is no open connection (contra Appx673 (Fig. 3)), and there is no means for *re-establishing* that connection (even if one had existed). Each map request is a *unique* request. In ArcView’s own words: “This means that ArcView must completely satisfy each MapCafe request without requiring that any of the user’s current state, such as which themes are currently turned on in MapCafe, be carried over to satisfy the next request.” Appx995. ArcView cannot anticipate without disclosing that element.

2. ArcView also fails because it discloses a unitary call, not a “communication connection.” Appx20. As explained above, ArcView is designed to transmit

individual, static, HTML calls to an intermediate server, which passes them along to another server (where the map is generated); it does not involve an open connection that permits interactive steps until the connection is terminated. Compare Appx994-995, with Appx673, 685 (Fig. 3, 13:38-40). These unitary calls do not establish the claimed “connection.”

In response, the Examiner argued that “communication connection” is a “broad term” and “clearly encompasses” ArcView’s activity. Appx20-21. And according to the Board, “the ’838 Patent does not provide a definition of ‘communication connection’ that would *exclude* the disclosure of ArcView as communication connection established between MapCafe and the host computer.” Appx21 (emphasis added).

This, again, misconstrues the claims and fails to understand the patent’s fundamental teachings. A static call to a server is not a communication connection; it does not open up the potential for seamless interaction, but instead calls for retrieving a single page in response to a single request. Appx995. That falls far short of the specification’s repeated description of “communication connection,” and it would fail to promote the invention’s core purpose: permitting the remote application to present itself (with all its functionality) on the client computer. See, *e.g.*, Appx673, 685 (Fig. 3a, 13:38-40 (“the droplets facilitate an interactive communi-

cation environment between the client computer 20 and the application server 40”).⁴

In any event, the Board’s rationale is entirely conclusory. It does not explain why ArcView teaches the necessary steps, but instead suggests that the patent does not *exclude* the Examiner’s view—which is not the same as an affirmative finding that the disclosure is sufficient. The APA requires the Board to provide an adequate, reasoned disposition of contested issues. It is not enough to state a conclusion—especially one that simply says the patent (for unknown reasons) does not necessarily *exclude* one reading, without any attempt to explain whether the non-excluded reading is reasonable or correct.

3. ArcView fails yet again because its “presentation information” is not based on the client computer’s “operating environment information.” Appx693 (29:60-63) (claim 1 requiring “sending *operating environment information regarding the client computer* from the client computer to the host computer,” and “retrieving presentation information to present an application and content, *the presen-*

⁴ Under the ’838 patent, the communication connection remains open until the interaction is terminated. Appx673, 685 (Fig. 3). This is fundamentally different from a series of distinct, unitary transactions, even if such transactions are between the same client and host. The patent expressly contemplates an *open* connection, where the application can operate remotely but seamlessly send and receive commands across the connection to replicate the feel and operation of a local application. That functionality is created by the patent’s use of client and host resources; it cannot be created or allowed by repeated static HTML calls to the same server.

tation information being based on the operating environment information”) (emphases added).

Nothing in ArcView meets these elements: ArcView works with Java Applets, and “a Java Applet, by its very nature, is not based on the operating environment information.” Appx20 (acknowledging Droplets’ argument). The Board has no response, which again is reversible error.

In any event, Droplets’ position is correct. A Java Applet is designed to be platform independent, and ArcView can generate its map images without knowing anything about the operating-environment information of the client computer. Compare Appx995, with Appx682-683 (8:63-67, 9:5-11) (“The information identifying the operating environment on the client computers 20 provides information to the application server 40 regarding the operating system and hardware capabilities of the particular client computer 20 that requested the droplet-enabled content 36. * * * [N]ot all client computers 20 are capable of presenting, for example, full color, high-resolution graphics. By providing the operating environment of the requesting client computer 20 to the application server 40, the application server 40 provides information 43 to present the requested applications 41 on the client computer 20.”).

In response, the Examiner (but not the Board) suggested that ArcView meets this requirement by generating a URL with “the width and height in pixels of the

map display area, which themes are currently turned on or off in the MapCafe Table of Contents, and the scale of the current map.” Appx6240. But that is information about the browser, not the *computer’s operating environment*. It simply dictates the URL data call to the static ArcView server—so it can produce an appropriate image of a map. Appx995. Neither the input nor the output has anything to do with the required elements of claim 1. That claim is designed to distinguish between things like “computer workstations,” “PalmPilots,” and “internet-enabled radio telephones.” Appx683 (9:1-4); compare also, *e.g.*, Appx683 (9:11-16) (detailing the type of information presented in response to inputs about the computer’s operating environment). ArcView, with its reliance on Java’s platform independence, does not satisfy the text or purpose of these critical elements.

D. ArcView And Orenshteyn Fail To Teach Multiple Aspects Of The Claimed Invention—And Orenshteyn Is Not Even Prior Art

According to the Board, the combination of ArcView and Orenshteyn render the challenged claims obvious under 35 U.S.C. 103(a). But as established above, ArcView fails to teach multiple elements of the challenged claims, and Orenshteyn is only used “to demonstrate the remote storage of a link for reconnecting to a host computer” (Appx21). That means this Court can reverse this rejection simply by relying on ArcView’s separate shortcomings. See Part I.C.2-I.C.3, *supra*.

In any event, Orenshteyn is not valid prior art and it does not teach the claimed element.

1. As Droplets established throughout the IPR proceeding, Orenshteyn is not valid prior art. See, *e.g.*, Appx5859; Appx6263; Appx4582-4584. The Examiner relied on a continuation-in-part that added significant new material to supplement the original parent. Droplets identified specific passages invoked in petitioners' claim charts (which the Examiner incorporated in its decision) that were found in the continuation-in-part but *not* the original parent. That new material greatly expanded critical concepts (*e.g.*, changing a “check point file” to “check point data”), and it likely affected the teaching and influence of the earlier parent—especially as relevant to the elements here. (This is presumably why that material was featured in petitioners' original claim charts.)

The Board's response to this showing was woefully incomplete: It simply declared, without analysis or explanation, that no new material was relied upon, without explaining why the old material alone was sufficient—and without parsing the claim elements and parent disclosures to show a necessary identity between elements. Appx22. This cursory explanation (to the extent it *explains* at all) is inadequate under the APA, and it alone discredits this rejection. See *Cutsforth*, 636 F. App'x at 578.

2. Orenshteyn anyhow falls short on the merits. According to the Board, Orenshteyn satisfies the claims (storing a “link” on the “client computer” for “re-establishing” the “communication connection”) by transmitting “selectable graphical icons” that may “spawn a corresponding application.” Appx22.

First, there is no indication that those “graphical icons” are *stored* on the computer, rather than merely being *displayed* while the window is open. Unless they are actually “stor[ed], on the client computer,” they fail to teach the claimed elements. See also Part I.A.2, *supra*.

Second, these graphical icons do not teach “re-establishing” a connection; at most, they teach how to *launch an application*. The Board simply conflates two distinct concepts: even if clicking an icon “spawns” an application, there is no indication that it also “re-establishes” a preexisting connection to a specific host computer. Showing that Orenshteyn provided a way to launch an application does not render obvious the vastly different approach of the claimed invention.

Third, the Board’s logic overlooks a critical point of *storing* the link. The user has to initiate contact with a server in Orenshteyn before a link *is presented*; this is far different from having a local link with functionality to automatically reestablish a past connection. See Appx941 (22:12-26).

Finally, the Board’s determination is entirely conclusory. It fails to cite any evidence to explain where or how the “link” is stored, or provide any reasoned

analysis justify its position. The Board is obligated to explain its views before invalidating an issued patent. Its failure to do so here warrants reversal. *Lee*, 277 F.3d at 1342, 1345.

II. THE BOARD'S INTER PARTES REEXAMINATION IS UNCONSTITUTIONAL UNDER ARTICLE III AND THE SEVENTH AMENDMENT

This IPR proceeding is unconstitutional under Article III and the Seventh Amendment, and the judgment below accordingly should be vacated. Under inter partes reexamination, third parties are permitted to litigate the validity of issued patents before Article I judges—even when (and perhaps especially when) the same parties are engaged in Article III litigation over the same patents. This effectively permits Article I judges to adjudicate private disputes that do not directly implicate public rights. It also creates the prospect, as here, of Article I judges extinguishing private-property rights without the protections and procedures of an Article III court.

Congress's decision to vest judicial power outside the judicial branch violates Article III, and its decision to permit agencies to extinguish vested property rights without a jury trial violates the Seventh Amendment. See, e.g., *Stern v. Marshall*, 564 U.S. 462, 484 (2011) (explaining the unconstitutionality of permitting state-law counterclaims to be tried in non-Article III bankruptcy proceedings); *Granfinanciera, S.A. v. Nordberg*, 492 U.S. 33, 41-42 (1989).

While Droplets acknowledges that this Court has held otherwise (see, *e.g.*, *MCM Portfolio LLC v. Hewlett-Packard Co.*, 812 F.3d 1284 (Fed. Cir. 2015); *Joy Techs. v. Manbeck*, 959 F.2d 226 (Fed. Cir. 1992)), it respectfully preserves this issue for further review.

CONCLUSION

The Board's judgment should be reversed and the case should be remanded with instructions to confirm the patentability of claims 6 and 20; in the alternative, the Board's judgment should be reversed and the case should be remanded with instructions to dismiss in light of the unconstitutionality of IPR review.

Respectfully submitted.

Timothy J. Bechen
BECHEN PLLC
9200 Forest Hill Ave., Suite 4-C
Richmond, VA 23235
Tel.: (804) 928-7037
tim@bechenlaw.com

/s/ Daniel L. Geyser
Daniel L. Geyser
Principal Attorney
STRIS & MAHER LLP
725 S. Figueroa Street, Suite 1830
Los Angeles, CA 90017
Tel.: (213) 995-6811
Fax: (213) 261-0299
daniel.geyser@strismaher.com

Counsel for Appellant Droplets, Inc.

December 9, 2016

TABLE OF CONTENTS

	Page
Board's Final Decision (March 30, 2016)	Appx1
U.S. Patent No. 7,502,838.....	Appx669



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/002,330	09/14/2012	7502838	2525.968REX2	7397
111420	7590	03/30/2016	EXAMINER	
Bechen PLLC 9200 Forest Hill Ave Suite 4-C Richmond, VA 23235			FERRIS III, FRED O	
			ART UNIT	PAPER NUMBER
			3992	
			MAIL DATE	DELIVERY MODE
			03/30/2016	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE INC. and FACEBOOK INC.
Requester

v.

DROPLETS, INC.
Patent Owner and Appellant

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2
Technology Center 3900

Before RICHARD M. LEOVITZ, JEFFREY B. ROBERTSON, and
ANDREW J. DILLON, *Administrative Patent Judges*.

DILLON, *Administrative Patent Judge*

DECISION ON APPEAL

STATEMENT OF THE CASE

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Patent Owner appeals under 35 U.S.C. § 134(b) (2002) from the final decision of the Examiner adverse to the patentability of claims 1–38. We have jurisdiction under 35 U.S.C. § 315 (2002).

We affirm.

Invention

The '838 patent describes a method and system for delivering interactive links for presenting applications and second information at a client computer from remote resources in a network-configured computer processing system. Abstract.

Claims

Claims 1–38 are subject to reexamination and have been rejected. Claims 1–38 are original patent claims. Claims 1, 15, 29, and 33 are independent.

Claim 1 is illustrative (emphasis added).

1. A method for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the method comprising:

retrieving, in response to a request of a client computer, a content item having computer program code embedded therein, execution of the embedded computer program code establishing a communication connection to a host computer;

sending operating environment information regarding the client computer from the client computer to the host computer;
retrieving presentation information to present an application and content, the presentation information being based on the operating environment information and comprising at least one of instructions

Appeal 2015-007929

Reexamination Control 95/002,330

Patent US 7,502,838 B2

for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the presented application; and

presenting, at the client computer, the application and the content based upon the presentation information.

Prior Art

Davis	US 5,796,952	Aug. 18, 1998
Frese	US 5,909,545	June. 1, 1999
Hickman	US 6,173,332 B1	Jan. 9, 2001
Orenshteyn	US 6,393,569 B1	May 21, 2002

ArcView Internet Map Server, Map Publishing on the Web, pp.1–60, Environmental Systems Research Institute, Inc., United States (1996–97) (hereinafter “ArcView”).

Livingston and Straub, Windows 98 Secrets, pp. 1–1207, Hungry Media, Inc., United States (March 24, 1998) (hereinafter “Livingston”).

The Santa Cruz Operation Technical White Paper, Tarantella—The Universal Application Server, pp.1–13, (July 1997) (hereinafter “Tarantella”).

Hahn, The Internet Complete Reference, Second Edition, pp. 1–161 and 176–199, Osborne McGraw-Hill, United States (1996) (hereinafter “Hahn”).

Neibauer, Running Microsoft Outlook 98, pp. 1–239, Microsoft Press, United States (1998) (hereinafter “Neibauer”).

Owner’s Contentions

Patent Owner contends that the Examiner erred in entering the following grounds of rejections against claims 1–38 (App. Br. 7–44; Reb. Br. 2–22):

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

A. The rejection of claims 1, 15, 29–36, and 38 under 35 U.S.C. § 102(b) as anticipated by Frese. App. Br. 7–11; Reb. Br. 17–18.

B. The rejection of claims 2, 4–8, 12–14, 16, 18–22, 26–28, 31, 32, and 37 under 35 U.S.C. § 103(a) as unpatentable over Frese in view of Hickman. App. Br. 11–15; Reb. Br. 2–5, 13–15.

C. The rejection of claims 3 and 17 under 35 U.S.C. § 103(a) as unpatentable in view of Frese, Hickman, and Admitted Prior Art (“APA”). App. Br. 15–16.

D. The rejection of claims 2, 3, 6, 9, 12, 16, 17, 20, 23, and 26 under 35 U.S.C. § 103(a) as unpatentable in view of Frese and Windows Secrets. App. Br. 16–19; Reb. Br. 5–6, 15.

E. The rejection of claims 9, 10, 23, and 24 under 35 U.S.C. § 103(a) as unpatentable over Frese in view of Hickman in view of Outlook 98. App. Br. 19–20.

F. The rejection of claims 11 and 25 under 35 U.S.C. § 103(a) as obvious in view of Frese, Hickman, Outlook 98, and Hahn. App. Br. 20–21.

G. The rejection of claims 1, 2, 4, 5, 7–9, 15, 16, 18, 19, 21–23, 29, 30, and 33–38 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella and Frese. App. Br. 21–31; Reb. Br. 6–8, 18–21.

H. The rejection of claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, and Windows Secrets. App. Br. 31–32.

I. The rejection of claims 2, 4, 5, 7, 8, 12–14, 16, 18, 19, 21, 22, and 26–28 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, and Orenshteyn. App. Br. 32–34; Reb. Br. 8–9.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

J. The rejection of claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of the combination of Tarantella, Frese, Orenshteyn, and Windows Secrets. App. Br. 34–35.

K. The rejection of claims 10, 11, 24, and 25 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, Orenshteyn, Windows 98, and Davis. App. Br. 35–36.

L. The rejection of claims 1, 2, 4–6, 15, 16, 18–20, 29, and 31–38 under 35 U.S.C. § 102(a) as anticipated by the published document ArcView. App. Br. 36–41; Reb. Br. 9–11, 15–16, 21–22.

M. The rejection of claims 2, 4–6, 16, and 18–20 under 35 U.S.C. § 103(a) as unpatentable in view of the combination of ArcView and Orenshteyn. App. Br. 41–42; Reb. Br. 11–12, 16–17.

N. The rejection of claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView, Orenshteyn, and Windows Secrets. App. Br. 42–43.

O. The rejection of claims 2, 12–14, 16, and 26–28 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView and Windows Secrets. App. Br. 43; Reb. Br. 12–13.

P. The rejection of claims 9, 10, 23, and 24 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView and Outlook 98. App. Br. 44.

Q. The rejection of claims 11 and 25 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView, Outlook 98, and Hahn. App. Br. 44

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

ANALYSIS

Ground of Rejection A

Patent Owner argues the Examiner erred in rejecting claims 1, 15, 29-36, and 38 under 35 U.S.C. § 102(b) as anticipated by Frese because Frese fails to disclose “sending operating environment . . . to the host computer.” Patent Owner argues that the Examiner asserted that remote application server (RAS) 20 in Frese depicts the host computer and that user system 16 depicts the client computer. Patent Owner points out that Frese requires remote control service publisher (RCSP) 12 to be the intermediary “because the RAS 20 operates to execute the application 22 and mirror the user system 16.” App. Br. 8. Consequently, Patent Owner argues that Frese teaches sending operating environment information to RSCP 12, which is not the host computer, but rather operates as an intermediary. App. Br. 7–9.

Patent Owner also argues that Frese fails to disclose “presenting, at the client computer, the application and the content based upon the presentation information.” Specifically, Patent Owner argues that Frese operates to insulate application 22 on the RAS 20 and consequently, the remote display module (RDM) cannot be properly interpreted as the claimed “application,” as relied upon by the Examiner. *Id.* at 10–11; Reb. Br. 17–18.

In response to Patent Owner’s first argument above, the Examiner notes that in one embodiment, at column 8, lines 5–8, Frese expressly teaches that RAS 20 “may reside on the same computer implementing RCSP 12” and consequently does disclose “sending operating environment” to the “host computer” in that embodiment. Ans. 9.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Regarding Patent Owner's second argument that the RDM "cannot be properly interpreted as the claimed 'application,'" the Examiner points out that Frese, at column 5, lines 26–29 expressly states "[b]ecause the RDM is identified as an application program for the user's computer, it is provided with its own communication access to the local resource interface for the user's computer." *Id.* at 13.

In view of the above cited express statements within the Frese reference, which we find support the exact interpretation adopted by the Examiner, Patent Owner's arguments with respect to Ground of Rejection A are unavailing. Accordingly, we affirm the Examiner's decision to reject the claims as anticipated by Frese.

Ground of Rejection B

Patent Owner argues the Examiner erred in rejecting claims 2, 4–8, 12–14, 16, 18–22, 26–28, 31, 32, and 37 under 35 U.S.C. § 103(a) as being unpatentable over Frese in view of Hickman.

Specifically, Patent Owner argues the cited references fail to show or suggest "storing on the client computer, a link for re-establishing the communication connection." Claim 2. The basis for Patent Owner's position is a belief that the URL disclosed by Hickman cannot be interpreted as the claimed "link" because a URL is a "location" and does not generate "some processor action," which Patent Owner asserts is an essential element of a "link." App. Br. 12–13; Reb. Br. 2–3.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Further, Patent Owner argues that Hickman fails to show or suggest the claimed “storing” the claimed link, but rather merely teaches that the client computer is “informed” of the URL. App. Br. 13–14.

Finally, Patent Owner argues that Hickman teaches the ability to access any of a number of network computers, and thus cannot be said to be “re-establishing **the communication connection to the host computer.**” *Id.* at 14–15; Reb. Br. 3–4.

With regard to Patent Owner’s arguments with respect to the alleged difference between a “link” and a “URL” the Examiner finds that Hickman teaches that in response to a request to connect to a “network accessible computer” (NAC), the URL of a NAC is provided, and by utilizing a Java Applet, converted to a “link,” as depicted at reference numeral 78 in Figure 3A of Hickman (“Click Me To Connect”). Ans. 14–15.

With respect to Patent Owner’s argument regarding “storing,” the Examiner finds that “inform[ing]” the computer in Hickman of the URL of the NAC necessarily requires that the subject URL be “stored” within that computer. *Id.* at 17–18.

Finally, in response to Patent Owner’s arguments regarding the existence of a number of network accessible computers (NACs) and the consequent failure of Hickman to teach “re-establishing” the link, the Examiner points out that Hickman, at column 4, lines 62–66, teaches that the described cluster “can be reduced to a single network-accessible computer” rendering each subsequent connection to the NAC a “re-establishing” of the connection. *Id.* at 19.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

We concur with the Examiner. We find the described “Click Me to Connect” disclosure within Hickman, in response to the provision of a URL, to be suggestive of the provision of a link. Further, we find “informing” a computer of any piece of data necessarily requires “storing” that data within the computer because storing would be necessary for the computer to receive the information. Finally, the express teaching of Hickman contemplates a cluster with a single network-accessible computer (NAC) rendering each subsequent connection a “re-establishing” of that connection.

Ground of Rejection C

Patent Owner argues the Examiner erred in rejecting claims 3 and 17 under 35 U.S.C. § 103(a) as unpatentable in view of Frese, Hickman, and Admitted Prior Art (“APA”) in view of the failure of the cited references to show or suggest a “drag and drop” operation with respect to the claimed “link.” App. Br. 15–16.

The Examiner finds that the URL of Hickman indeed suggests the claimed “link” and consequently, a “drag and drop” operation on the Hickman URL suggests the claimed feature.

We are not persuaded by Patent Owner’s arguments. As we noted above, we find the provision of a URL by Hickman to be equivalent to the claimed “link.” because Hickman teaches the URL is converted to a “link” by means of a Java Applet . Further, we acknowledge that “drag and drop” operations are notorious in the art and find that a graphic indication of a URL may be subject to a “drag and drop” operation as claimed.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Ground of Rejection D

Patent Owner argues the Examiner erred in rejecting claims 2, 3, 6, 9, 12, 16, 17, 20, 23, and 26 under 35 U.S.C. § 103(a) as unpatentable in view of Frese and Windows Secrets.

Patent Owner argues error on the basis of the previously argued distinction between a “link” and a “URL.” App. Br. 16–17; Reb. Br. 15.

Patent Owner also argues that the “**stateless nature of Frese**” prohibits any “link for re-establishing the communication connection.” Patent Owner points out that Frese does not use client state information and teaches away from the use of client state information and cannot therefore “re-establish” a communication connection. *Id.* at 18–19.

Finally, Patent Owner argues that claims 6 and 20, which depend from claims 2 and 16, recite “storing further comprises storing instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.” Patent Owner argues the claimed storing takes place on the client computer and that is not shown by the cited references. *Id.* at 19.

We find no patentable distinction between the claimed “link” and the disclosed “URL,” especially in view of the express description within Windows Secrets of the creation of “link” and the description of that “link” as a “URL shortcut.” Windows Secrets, p. 244.

The Examiner finds that Frese does not expressly prohibit any type of communication connection from being “re-established” and Patent Owner’s

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

assertions regarding the intentions of Frese are conclusory in nature. Ans. 25.

We agree with the Examiner. Aside from mere conclusions regarding reasons why Frese will not permit “re-establishing communication” Patent Owner has not pointed to any express basis for asserting that Frese prohibits the re-establishing of a communication connection.

With regard to the “storing” argument, the Examiner notes that although the cited portion of Frese describes functions which occur at the host, the data utilized to perform those functions is all stored at the client, as set forth in Frese at column 7, lines 40–62. When Frese is considered as a whole, the Examiner believes that the data stored at the client, even if utilized at the host, suggests the claimed “storing.” Ans. 26-27.

We find the Examiner’s position persuasive. While the host within Frese actually performs the various functions, that underlying data is stored within the client (as noted above) and thus meets the broadest reasonable interpretation of claims 6 and 20.

Ground of Rejection E

Patent Owner argues the Examiner erred in rejecting claims 9, 10, 23, and 24 under 35 U.S.C. § 103(a) as unpatentable over Frese in view of Hickman in view of Outlook 98.

Patent Owner urges error on the part of the Examiner based upon the failure of the cited references to disclose a “link.”

For the same reasons we set forth above, we do not find Patent Owner’s arguments persuasive.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Ground of Rejection F

Patent Owner argues the Examiner erred in rejecting claims 11 and 25 under 35 U.S.C. § 103(a) as unpatentable in view of Frese, Hickman, Outlook 98, and Hahn.

Specifically, Patent Owner argues the cited references fail to show or suggest “the information to track transmission of the link comprises a global unique identifier that is assigned to the link and information for identifying a given client computer that has received the link” as recited in claim 11. Patent Owner urges that the Examiner’s reliance on the URL as a global unique identifier assigned to the link, and the email address as information identifying the client computer are erroneous because a URL is unique only to the server and the email address does not teach that a given client computer has received a link. App. Br. 20–21.

The Examiner finds that Hahn expressly discloses “every hypertext item on the Net has an address of its own . . . [W]ithin this system, there is a unique URL for every hypertext item on the Net.” The Examiner interprets this unique URL as a globally-unique identifier. (*See* Hahn, p.177) Further, the Examiner finds that Outlook 98 teaches that an email address can be used to identify a given client computer that received a particular link. Ans. 29–31.

We find the Examiner’s interpretation to be reasonable and factually supported. Those skilled in the art will appreciate that a URL is indeed a globally-unique identifier for the reasons given by the Examiner. We find no more limiting definition within Patent Owner’s Specification. Similarly,

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

an email address, which is unique, can be utilized to identify a recipient of a link.

Ground of Rejection G

Patent Owner argues the Examiner erred in rejecting claims 1, 2, 4, 5, 7–9, 15, 16, 18, 19, 21–23, 29, 30, and 33–38 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella and Frese.

Patent Owner argues that Tarantella is not a valid prior art reference because it “**fails to include a technical description.**” Specifically, Patent Owner urges that Tarantella is an “informal” document and fails as an enabling disclosure. Patent Owner argues that Tarantella includes only a general description of an Adaptive Internet Protocol and was incomplete until the issuance of U.S. Patent No. 6,104,392, filed on November 12, 1998. App. Br. 21–26.

Further, Patent Owner argues that Tarantella fails to disclose the claimed “operating environment information regarding the client computer,” “sending operating environment information regarding the client computer to the host computer,” the “presentation information being based on the operating environment information,” and “presenting, at the client computer, the application and the content based upon the presentation information.” *Id.* at 26–30.

Finally, Patent Owner argues that Frese fails to cure the argued deficiencies of Tarantella and that Frese and Tarantella together fail to suggest the claimed “link for re-establishing” for the reasons set forth above. App. Br. 30–31.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

First, the Examiner finds that Tarantella was not cited under 35 U.S.C. § 102, but rather under § 103 and is considered prior art for all that it teaches. Ans. 33. With respect to Patent Owner's argument that a subsequent issued patent based on Tarantella was not enabling, the Examiner again emphasizes that the Tarantella publication is being relied upon for its teachings. *Id.* at 34. With regard to Patent Owner's argument that Tarantella fails to disclose sending operating environment information regarding the client computer to the host computer, the Examiner points out that Tarantella, at page 12, recites "[t]he first phase in this link is to pass parameters identifying the characteristics of the client device and network connection" which is utilized to display the application on both the client and AIP monitors, which the Examiner interprets as operating environment/presentation information which is sent to the host. *Id.* at 36-37, 39, 40-41, and 42.

With regard to Patent Owner's arguments concerning the storage of a "link for re-establishing" the Examiner points out that Tarantella, at page 10, teaches that an "application session can be configured, by administrators, to be resumable. In this scenario users can disconnect themselves from the Tarantella server but leave resumable applications running." The Examiner further finds that the "saved" application can be resumed by "clicking" on an icon, which the Examiner interprets as a "link." Ans. 44-45.

We agree with the Examiner. The Tarantella reference, despite Patent Owner's characterization as lacking "technical description" is still a valid reference for what it teaches. Further, we find that the "characteristics of the client device and network connection" in Tarantella to be a disclosure of

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

passing parameters, which identify the characteristics of the client device and network connection. Finally, we find a “resumable” connection, as taught by Tarantella, to be a “link for re-establishing” a connection.

Ground of Rejection H

Patent Owner argues the Examiner erred in rejecting claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, and Windows Secrets.

The basis for the alleged error is, once again, the alleged failure of the cited references to show or suggest the “link for re-establishing the communication connection” for the reasons set forth above. App. Br. 31–32.

We find no error in the Examiner’s reliance on the teaching of a URL to suggest the claimed “link” for the same reasons set forth above.

Ground of Rejection I

Patent Owner argues the Examiner erred in rejecting claims 2, 4, 5, 7, 8, 12–14, 16, 18, 19, 21, 22, and 26–28 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, and Orenshteyn.

Specifically, Patent Owner argues this combination of references fails to show or suggest “storing, on the client computer, a link for re-establishing” as set forth above. App. Br. 32.

Further, Patent Owner argues that Orenshteyn is not a valid prior art reference, urging that the Examiner has relied upon portions of Orenshteyn that were not present in the parent application. *Id.* at 33–34.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Finally, Patent Owner argues that the Examiner failed to provide any actual support for the rejection of claims 21, 22, and 26–28. *Id.* at 34.

The Examiner notes that Tarantella, rather than Orenshteyn, is relied upon for the storage of a link to re-establish a communication connection, reiterating that Tarantella expressly states that applications are designed to be resumable, upon the selection of an icon. Ans. 47.

With respect to Patent Owner’s arguments regarding whether or not Orenshteyn is prior art, the Examiner acknowledges that Orenshteyn is a continuation-in-part, but asserts that those portions of Orenshteyn which were relied upon by the Examiner, exclude the new matter (Figures 4–8 and column 23, line 31 through column 25, line 37) added at filing of Orenshteyn. *Id.* at 49.

Finally, regarding the rejection of claims 20, 21, and 26–28, the Examiner acknowledges that these claims were included only by virtue of a typographical error. *Id.* at 50.

We find that the Examiner has made a clear showing that Tarantella clearly demonstrates a link (selectable icon) that, upon activation, re-establishes a connection with the application. Consequently, we find no error in the Examiner’s reliance on this combination of references.

Regarding the status of Orenshteyn, the Examiner has only relied upon those portions of the reference that are entitled to the filing date of the parent application. Ans. 49. We therefore find no merit in Patent Owner’s argument.

Finally, in view of the Examiner’s admission, we find that claims 20, 21, and 26–28 are not rejected over this cited combination of references;

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

however, we note that those claims were rejected under Ground of Rejection B.

Ground of Rejection J

Patent Owner argues the Examiner erred in rejecting claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of the combination of Tarantella, Frese, Orenshteyn, and Windows Secrets.

The basis for the alleged error is, once again, the alleged failure of the cited references to show or suggest the “link for re-establishing the communication connection” for the reasons set forth above. App. Br. 34–35.

We find no error in the Examiner’s reliance on the teaching of a URL to suggest the claimed “link” for the same reasons we set forth above.

Ground of Rejection K

Patent Owner argues the Examiner erred in rejecting claims 10, 11, 24, and 25 under 35 U.S.C. § 103(a) as unpatentable in view of Tarantella, Frese, Orenshteyn, Windows 98, and Davis.

Patent Owner argues that Davis fails to teach or suggest transmissions between clients, and cannot therefore suggest the monitoring of inter-client communications. App. Br. 35–36.

Additionally, Patent Owner argues that Davis does not show or suggest a **“global unique identifier that is assigned to the link and information for identifying a given client computer that has received the link.”** Patent Owner urges that Davis discloses two databases which are utilized to track browser activities. One database tracks user activities and the second database tracks web address activity. Patent Owner argues such

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

activity is inconsistent with the claimed identification of clients receiving a link. *Id.* at 36.

In response, the Examiner finds that Patent Owner only argues the Davis reference, while the Examiner's rejection is based upon a combination of references. The Examiner notes that transmissions of links between clients is taught by the combination of Tarantella, Frese, Orenshteyn, and Windows Secrets, while Davis is relied upon merely to show the storing of information that tracks those transmissions. Consequently, the Examiner urges that Patent Owner's argument that Davis fails to show tracking transmissions of links is not persuasive. Ans. 53.

One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *See In re Keller*, 642 F.2d 413, 426 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986). Here the Patent Owner argues that Davis fails to show or suggest a global identifier assigned to a link and information for identifying a client computer which received that information. The Examiner finds that the transmission of links between clients is taught by the combination of Tarantella, Frese, Orenshteyn and Windows Secrets, while Davis is merely relied upon for a teaching that the information is stored to track that transmission. *Id.* We therefore find no error in the Examiner's position for the reasons set forth above.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Ground of Rejection L

Patent Owner argues the Examiner erred in rejecting claims 1, 2, 4–6, 15, 16, 18–20, 29, and 31–38 under 35 U.S.C. § 102(a) as anticipated by the published document ArcView.

Patent Owner argues that ArcView fails to disclose “execution of the embedded computer program code establishing a communication connection to a host computer.” Specifically, Patent Owner argues that ArcView discloses a unitary call to the web server, which Patent Owner argues is not a communication connection and consequently, ArcView cannot be interpreted as disclosing the claimed invention. App. Br. 38–39.

Patent Owner also argues ArcView fails to disclose “presenting, at the client computer, the application and the content based upon the presentation information.” Patent Owner argues that the web browser application of ArcView is the MapCafe, but Patent Owner argues that downloading a Java Applet, by its very nature, is not based on the operating environment information, as required by claim 1. *Id.* at 40.

The Examiner finds that in at least one embodiment, disclosed at page 5 of ArcView, the Web Server resides in the same computer as the ArcView, as can thus be interpreted as the host, and therefore Patent Owner’s argument that the web browser in ArcView is entirely unaware of the existence of the ArcView server is without merit. Ans. 55.

Regarding Patent Owner’s arguments about the establishing of a communication connection, the Examiner finds that “communication connection” is a broad term and clearly encompasses the execution of

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

MapCafe which establishes a communication connection with the host as depicted at page 34 of ArcView. *Id.* at 56–58.

We agree with the Examiner. In a single computer embodiment, where the Web Server resides in the same computer as ArcView, we find the embedded computer program establishes a communication connection with the host computer. In addition, the '838 Patent does not provide a definition of “communication connection” that would exclude the disclosure of ArcView as communication connection established between MapCafe and the host computer.

Ground of Rejection M

Patent Owner argues the Examiner erred in rejecting claims 2, 4–6, 16, and 18–20 under 35 U.S.C. § 103(a) as unpatentable in view of the combination of ArcView and Orenshteyn.

Patent Owner argues that the stateless operation of ArcView teaches away from a combination with Orenshteyn, since the ArcView server must be available for receipt of map display requests while Orenshteyn requires a direct client/server connection. App. Br. 41.

Patent Owner also argues Orenshteyn fails to show or suggest storing a “link” for re-establishing a communication connection and, as argued above, is not a valid prior art reference. *Id.* at 42.

The Examiner finds that the proper test for a cited combination of references is not whether or not the references may be physically combined, but what the combination would suggest to one of ordinary skill in that art.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

In the present matter the Examiner points out that Orenshteyn was cited merely to demonstrate the remote storage of a link for reconnecting to a host computer, a feature absent from ArcView.

The Examiner points out that Orenshteyn discloses the use of a graphical icon, which when selected can be utilized to “spawn” an application, which the Examiner interprets as re-establishing communications.

Finally, the Examiner reiterates that only those portions of Orenshteyn which were entitled to the earlier filing date of the parent application were relied upon by the Examiner.

The Examiner is correct that physical combination of the elements described in two references is not required, and that the proper test for such a combination is the suggestion that combination would raise in the mind of one have ordinary skill in the art. Consequently, it is our opinion that the teaching of a stored link for re-establishing a communication connection, taught by Orenshteyn, as noted above, in combination with the teaching of ArcView would reasonably suggest the claimed limitations for the reasons given by the Examiner. .

Further, we find that the transmission of selectable graphical icons, as taught by Orenshteyn, which may be utilized to spawn a corresponding application, suggests the re-establishing of a communication connection, as claimed. Additionally, those portions of Orenshteyn cited by the Examiner are those portions having a proper basis in the parent application, rendering Orenshteyn a valid prior art citation.

We therefore find no error by the Examiner.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Ground of Rejection N

Patent Owner argues the Examiner erred in rejecting claims 3, 9, 17, and 23 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView, Orenshteyn, and Windows Secrets.

The basis for the alleged error is, once again, the alleged failure of the cited references to show or suggest the “link for re-establishing the communication connection” for the reasons set forth above. App. Br. 43.

We find no error in the Examiner’s reliance on the teaching of a URL to suggest the claimed “link” for the same reasons we set forth above.

Ground of Rejection O

Patent Owner argues the Examiner erred in rejecting 2, 12–14, 16, and 26–28 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView and Windows Secrets.

Patent Owner argues that the stateless operation of ArcView teaches away from a combination with Windows Secrets for the reasons set forth above in Ground of Rejection M. App. Br. 43.

Patent Owner also argues that Windows Secrets fails to teach or suggest a “link” as claimed, disputing the Examiner’s reliance on the teaching of a “URL.” *Id.*

For the same reasons we set forth above we find no basis for Patent Owner’s argument that ArcView cannot be physically combined with the Windows Secrets reference. As we noted above, physical combinability is not the proper test.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Further, we have repeatedly found that the Examiner's reliance on a "URL" as suggestive of a "link" is within the broadest reasonable interpretation of Patent Owner's claims. We therefore find no error on the part of the Examiner.

Ground of Rejection P

Patent Owner argues the Examiner erred in rejecting claims 9, 10, 23, and 24 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView and Outlook 98.

Patent Owner argues that ArcView fails to teach the claimed "link" and teaches away from such a limitation. Further, Patent Owner argues that the Examiner erred in relying upon Outlook 98 to cure that deficiency. App. Br. 44.

The Examiner notes that physical combination of the elements of references is not required and also finds that Outlook 98 expressly teaches transmitting a URL or a "link" within a message, citing pages 135 and 202.

In view of the fact that physical combination is not required, and in the absence of a clear teaching that expressly prohibits the combination of a reference, we find no error on the part of the Examiner. This is particularly true in view of the express teaching within Outlook 98 that a message may include a "link." See Outlook 98, pp. 135 and 202.

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

Ground of Rejection Q

Patent Owner argues the Examiner erred in rejecting claims 11 and 25 under 35 U.S.C. § 103(a) as unpatentable in view of ArcView, Outlook 98 and Hahn.

Specifically, Patent Owner argues the Hahn teaching of an email address does not teach or suggest a “unique identifier” assigned to a link. App. Br. 44.

The Examiner finds that under a broadest reasonable interpretation, Hahn discloses that a unique e-mail address as a global unique identifier and that each hypertext item on the internet has a unique address. Ans. 71.

In the absence of a narrow definition which would restrict the claim terms, we find no error in the Examiner’s rejection of claims 11 and 25 over the cited references.

Summary/Conclusion

We sustain the Examiner’s rejections of claims 1–38.

DECISION

The Examiner’s decision adverse to the patentability of claims 1–38 is affirmed.

Requests for extensions of time in this proceeding are governed by 37 C.F.R. §§ 1.956 and 41.79(e).

AFFIRMED

Appeal 2015-007929
Reexamination Control 95/002,330
Patent US 7,502,838 B2

PATENT OWNER:

BECHEN PLLC
9200 Forest Hill Ave
Suite 4-C
Richmond, VA 23235

THIRD PARTY REQUESTER:

Michael V. Messinger
STERNE KESSLER GOLDSTEIN & FOX P.L.L.C
1100 New York Ave., N.W.
Washington, DC 20005

US007502838B2

(12) **United States Patent**
Franco et al.(10) **Patent No.:** **US 7,502,838 B2**
(45) **Date of Patent:** ***Mar. 10, 2009**(54) **SYSTEM AND METHOD FOR DELIVERING REMOTELY STORED APPLICATIONS AND INFORMATION**

FOREIGN PATENT DOCUMENTS

EP	199964264	7/1997
WO	0120848	7/2007

OTHER PUBLICATIONS

Richard S. Hall, et al. (1997), "An Architecture for Post-Development Configuration Management in A Wide-Area Network", Software Engineering Research Laboratory, Department of Computer Science, Proceeding of the 1997 Inter. Conf. on Distributed Computing Systems, Baltimore Maryland, pp. 269-278.

(Continued)

Primary Examiner—David Y Eng(74) *Attorney, Agent, or Firm*—Seth Ostrow, Ostrow Kaufman & Frank LLP(57) **ABSTRACT**

A method and system are disclosed for delivering interactive links for presenting applications and second information at a client computer from remote sources in a network-configured computer processing system. In one embodiment, the method includes retrieving over a first communication connection, in response to a request of a client computer, informational content having computer program code embedded therein, and executing the embedded computer program code for establishing a second communication connection to an application server. The method further includes retrieving over the second communication connection first information including presentational information for presenting the application and the second information. The method also includes presenting the application and the second information based upon the presentational information, and storing on the client computer an interactive link for selectively re-establishing the second communication connection to the application server for retrieving the first information and presenting the application and the second information on an as-needed basis. Preferably, the storing of the interactive link includes downloading a graphical representation of the interactive link and storing a file containing information representing an operating environment of the client computer and a network address of the application server.

(75) **Inventors:** Louis M. Franco, Hoboken, NJ (US); Frank Leon Rose, Brooklyn, NY (US); Philip S. J. Brittan, Brooklyn, NY (US); Mark Cunningham, Brooklyn, NY (US); Alex Bulkin, Brooklyn, NY (US); Mat Baskin, Bronx, NY (US); Greg Blonder, Summit, NJ (US)

(73) **Assignee:** Droplets, Inc., Dallas, TX (US)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 76 days.

This patent is subject to a terminal disclaimer.

(21) **Appl. No.:** 10/720,728(22) **Filed:** Nov. 24, 2003(65) **Prior Publication Data**

US 2004/0199574 A1 Oct. 7, 2004

Related U.S. Application Data

(63) Continuation of application No. 09/599,382, filed on Jun. 22, 2000, now Pat. No. 6,687,745.

(60) Provisional application No. 60/153,917, filed on Sep. 14, 1999.

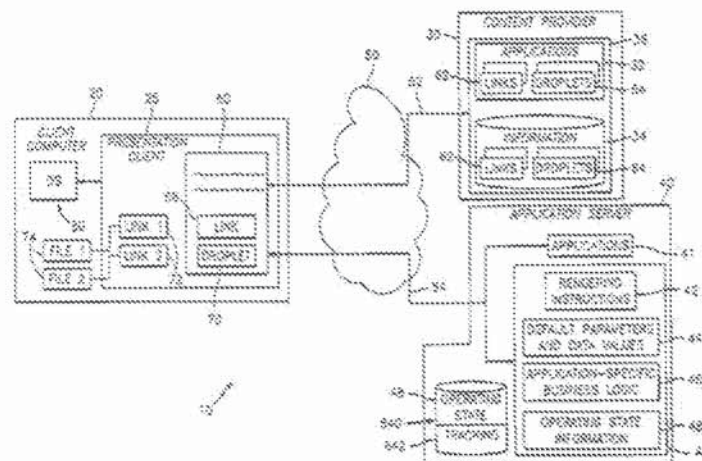
(51) **Int. Cl.**
G06F 15/16 (2006.01)(52) **U.S. CL.** 709/219(58) **Field of Classification Search** 709/203, 709/202, 213, 217

See application file for complete search history.

(56) **References Cited****U.S. PATENT DOCUMENTS**

5,204,247 A 4/1993 Bernstein et al. 395/157

(Continued)

38 Claims, 8 Drawing Sheets

US 7,502,838 B2

Page 2

U.S. PATENT DOCUMENTS

5,276,607 A	1/1994	Harris	6,553,417 B1	4/2003	Gampper	709/224
5,717,939 A	2/1998	Bricklin	6,636,885 B1	10/2003	Martin	709/202
5,742,835 A	4/1998	Kaethler	6,654,814 B1	11/2003	Britton	
5,767,847 A	6/1998	Mori	6,687,745 B1	2/2004	Franco	
5,768,158 A	6/1998	Adler	6,691,157 B2	2/2004	Muir	709/217
5,768,510 A	6/1998	Gish	6,697,861 B2	2/2004	Barnier et al.	709/223
5,796,952 A	8/1998	Davis	6,782,431 B1	8/2004	Mukherjee et al.	709/246
5,802,530 A	9/1998	Van Hoff	6,799,195 B1 *	9/2004	Thibault et al.	709/203
5,805,815 A *	9/1998	Hill	6,810,410 B1	10/2004	Durham	709/203
5,838,906 A	11/1998	Doyle et al.	7,206,085 B1 *	4/2007	Eguchi	358/1.15
5,838,910 A	11/1998	Domenikos et al.	7,243,352 B2 *	7/2007	Mandava et al.	718/104
5,838,916 A *	11/1998	Domenikos et al.	7,321,917 B2	1/2008	Durham	
5,848,246 A	12/1998	Gish	7,340,716 B1	3/2008	Chandler	
5,877,765 A *	3/1999	Dickman et al.	2001/0644819 A1	11/2001	Gong	
5,877,767 A	3/1999	Yohanan	2002/0095478 A1	7/2002	Muir et al.	709/217
5,889,942 A	3/1999	Orenshteyn	2003/0018885 A1	1/2003	Landsman et al.	713/2
5,890,161 A *	3/1999	Holland et al.	2003/0154261 A1	8/2003	Doyle et al.	709/218
5,903,901 A *	5/1999	Kawakura et al.	2005/0010647 A1	1/2005	Durham	709/212
5,905,248 A	5/1999	Russell et al.				
5,915,112 A *	6/1999	Boutcher				
5,919,247 A	7/1999	Van Hoff				
5,933,599 A *	8/1999	Nolan				
5,940,074 A *	8/1999	Brit et al.				
5,950,608 A	9/1999	van Hoff				
5,966,705 A	10/1999	Koneru				
5,970,506 A	10/1999	Kiyan				
5,987,245 A	11/1999	Gish				
5,999,972 A	12/1999	Gish				
6,009,410 A *	12/1999	LeMole et al.				
6,011,918 A	1/2000	Cohen				
6,026,437 A	2/2000	Muscheft				
6,029,141 A	2/2000	Bezos				
6,035,324 A *	3/2000	Chang et al.				
6,038,590 A	3/2000	Gish				
6,043,815 A	3/2000	Simonoff et al.				
6,052,711 A	4/2000	Gish				
6,064,382 A	5/2000	Diedrich et al.				
6,065,943 A *	5/2000	Domenikos et al.				
6,076,166 A *	6/2000	Moshfeghi et al.				
6,078,321 A	6/2000	Simonoff et al.				
6,078,322 A	6/2000	Simonoff et al.				
6,088,515 A	7/2000	Muir et al.				
6,091,412 A	7/2000	Simonoff et al.				
6,105,063 A	8/2000	Hayes				
6,115,040 A	9/2000	Blades et al.				
6,115,741 A *	9/2000	Domenikos et al.				
6,226,654 B1	5/2001	Van Hoff				
6,233,620 B1	5/2001	Gish				
6,240,442 B1	5/2001	Domenikos et al.				
6,253,282 B1	6/2001	Gish				
6,266,709 B1	7/2001	Gish				
6,272,555 B1	8/2001	Gish				
6,272,556 B1	8/2001	Gish				
6,304,893 B1	10/2001	Gish				
6,317,761 B1	11/2001	Landsman et al.				
6,324,578 B1	11/2001	Cox				
6,331,746 B1	2/2002	Gebauer				
6,357,938 B1	3/2002	Scouten				
6,362,836 B1	3/2002	Shaw et al.				
6,362,840 B1	3/2002	Burg et al.				
6,370,552 B1	4/2002	Bloomfield				
6,370,570 B1	4/2002	Muir et al.				
6,393,569 B1	5/2002	Orenshteyn				
6,424,991 B1	7/2002	Gish				
6,434,598 B1	8/2002	Gish				
6,460,123 B1	10/2002	Blumenau				
6,463,343 B1	10/2002	Emens et al.				
6,470,378 B1	10/2002	Tracton				
6,535,912 B1	3/2003	Anupam et al.				
6,544,295 B1	4/2003	Bodnar				
6,546,554 B1	4/2003	Schmidt				

OTHER PUBLICATIONS

The Castanet Tuner and How Channels Appear, 2 pages, Jun. 12, 2007.

Lemay, Official Guide to Castanet, Sams.net., 1997, 195 pages.

pcAnywhere User's Guide v9.2, Symantec, 1999.

URL: <http://web.archive.org/web/20001110010400/java.sun.com/features/2000>, Wang (2000) Announcing Java web Start Technology, 5 pages.

Vanderheiden and Chisholm, Central Reference Document - Version 8 Unified Web Site Accessibility Guidelines, Jan. 20, 1998.

W3C Working Draft, Techniques for Web Content Accessibility Guidelines, Mar. 24, 1999.

W3C Recommendation, Web Content Accessibility Guidelines 1.0, May 5, 1999.

W3C Note, Composite Capability/Preference Profiles (CC/PP): A User Side Framework for Content Negotiation, Jul. 27, 1999.

W3C Note, Composite Capability/Preference Profiles (CC/PP): A User Side Framework for Content Negotiation, Nov. 30, 1998.

W3C Note, Client-Specific Web Services by Using User Agent Attributes, Dec. 30, 1997.

Hollman and Minz, RFC 2295, Transparent Content Negotiation in HTTP, Mar. 1998.

Fielding et al., RFC 2068, Hypertext Transfer Protocol - HTTP/1.1, Jan. 1997.

Fielding et al., RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, Jun. 1999.

Seshan, Benefits of Transparent Content Negotiation in HTTP, Nov. 1998.

Castanet system, Marimba, no later than Jan. 11, 1998.

HotJava Views, Sun Microsystems, 1997.

ICE-T system, Sun Microsystems, no later than 1996.

Fielding et al., RFC 1945, Hypertext Transfer Protocol - HTTP/1.0, May 1996.

Marimba Screen Shots, 2 pages.

The Castanet Solution, 11 pages.

Java Network Launcher Protocol & API Specification, Version 1.0, JSR-56 (2000) Sun Microsystems, 75 pages.

URL: <http://www.sun.com/smi/press/sunflash/2000-06/sunflash.20000606.3.xml>, Sun Microsystems Delivers next Generation Client-Side Java Technology, 3 pages.

Knut Manske, et al., (1998) "Comic Actors Representing Software Agents", Teleoperation Department, University of Linz, Austria, IEEE proceedings 1998, pp. 213-222.

Supplemental European Search Report dated Sep. 14, 2006; and

International Search Report dated Jan. 9, 2001.

* cited by examiner

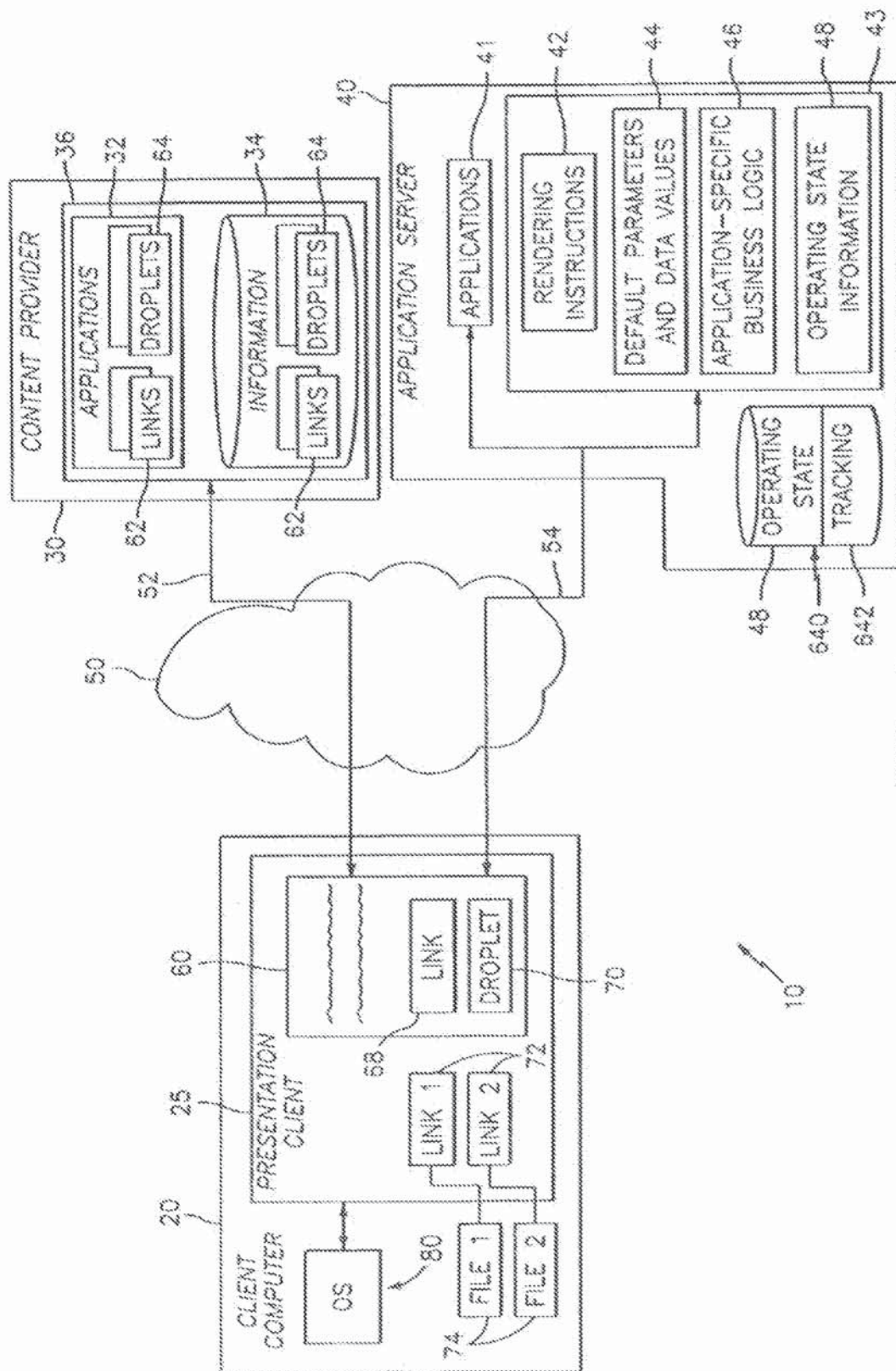


FIG. 1

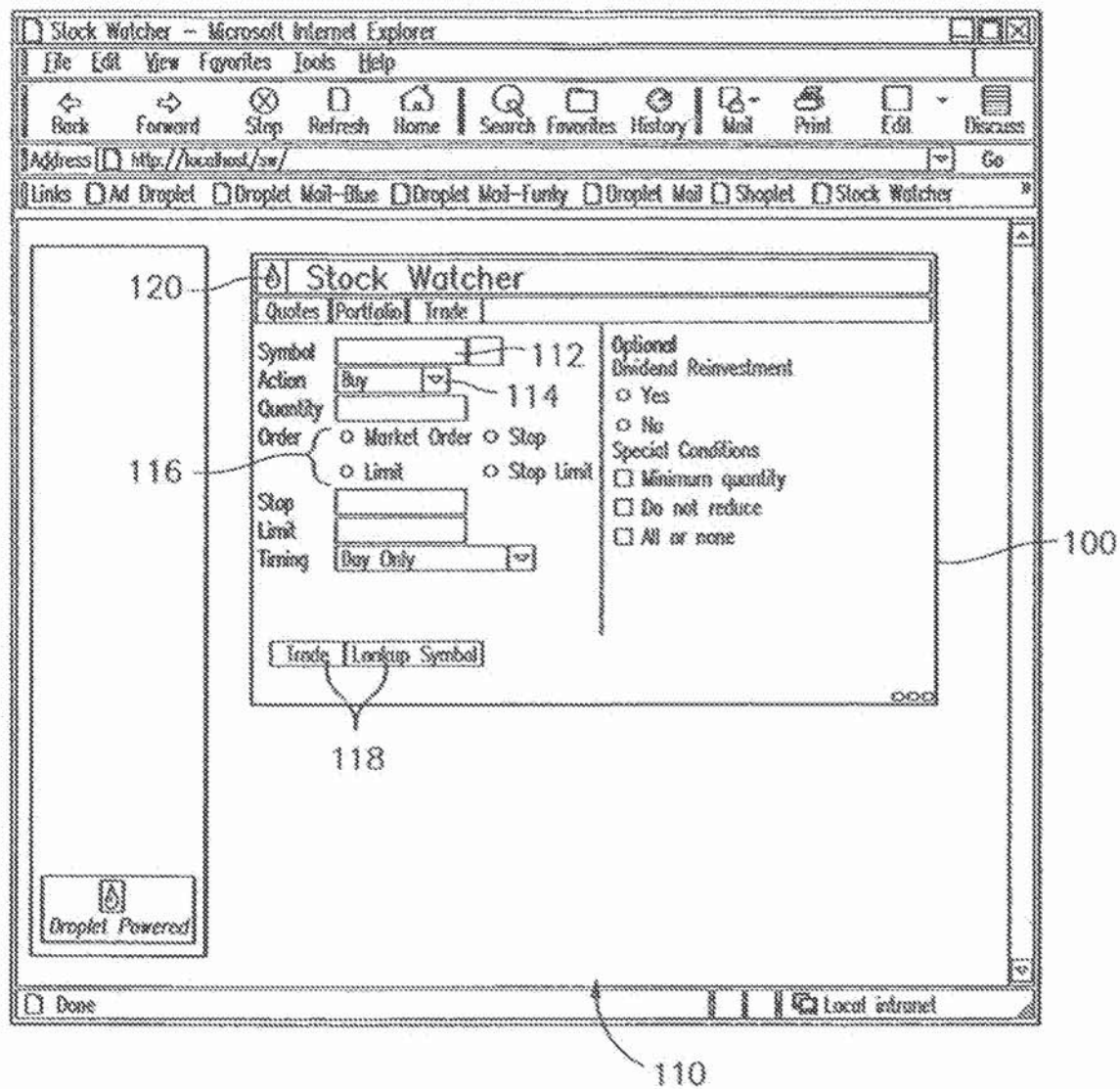


FIG. 2

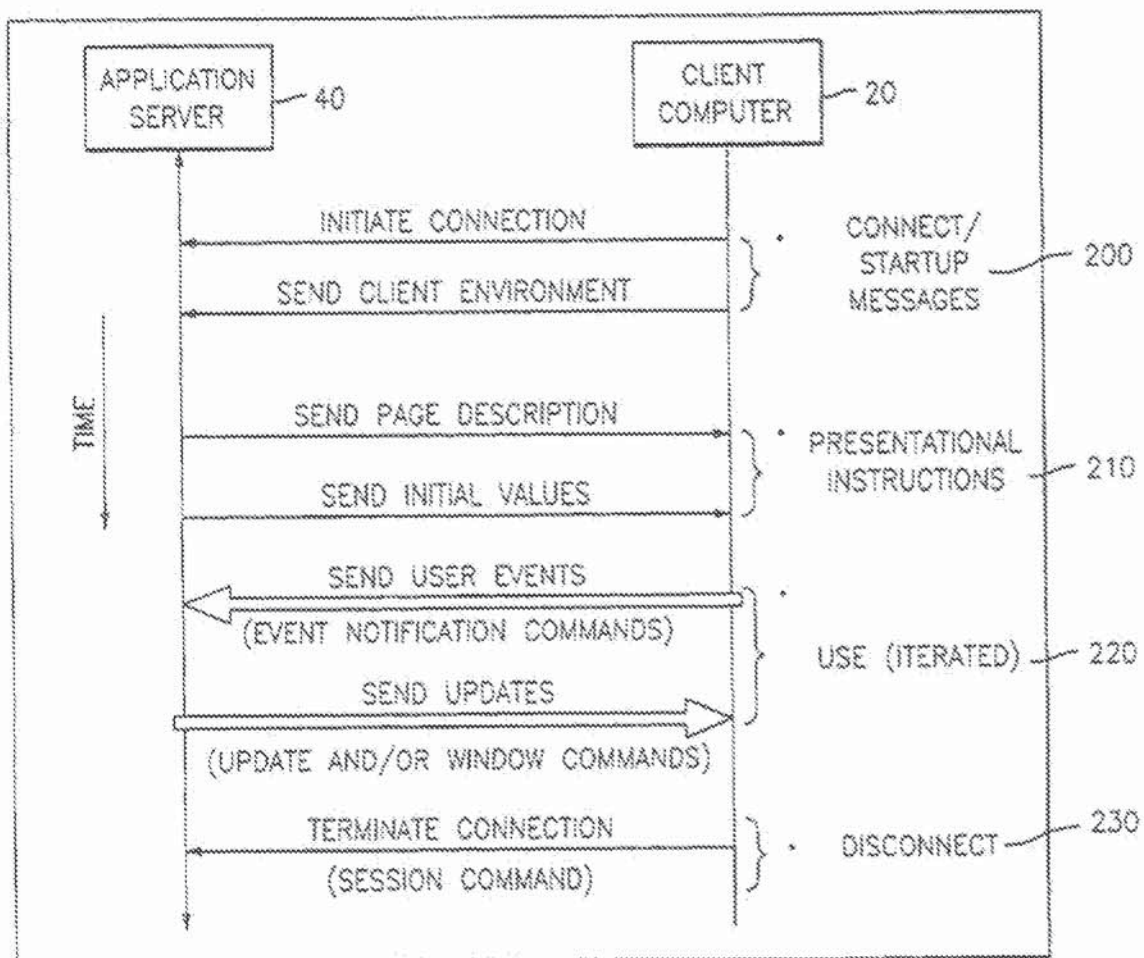


FIG. 3

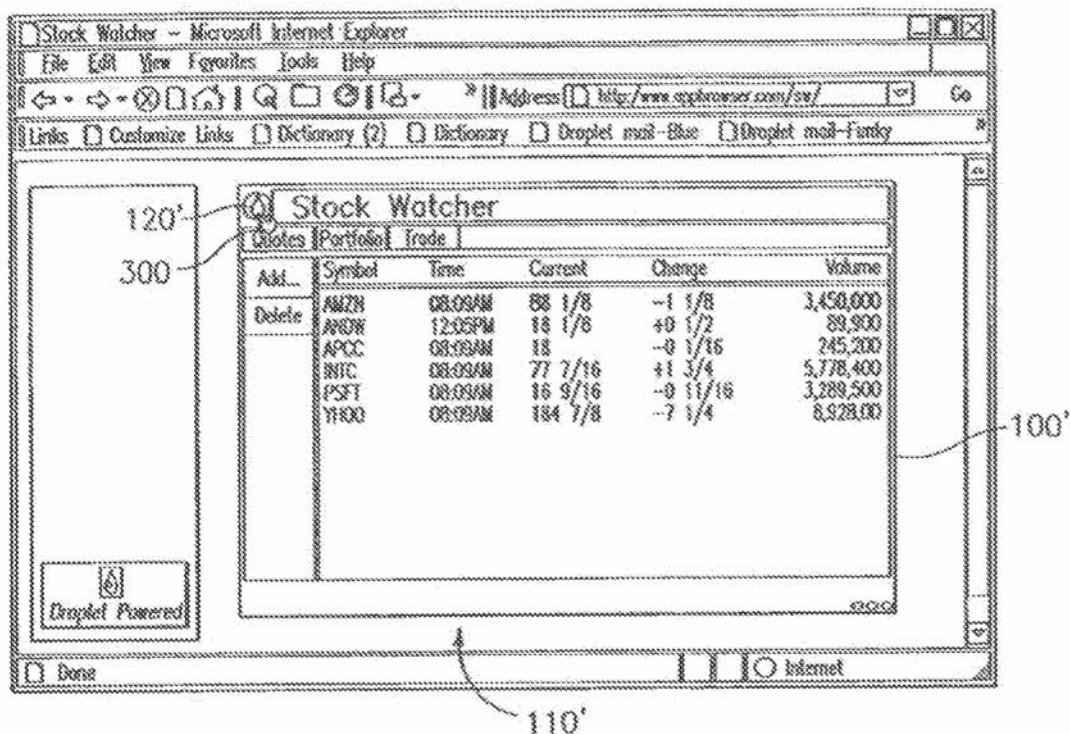


FIG. 4A

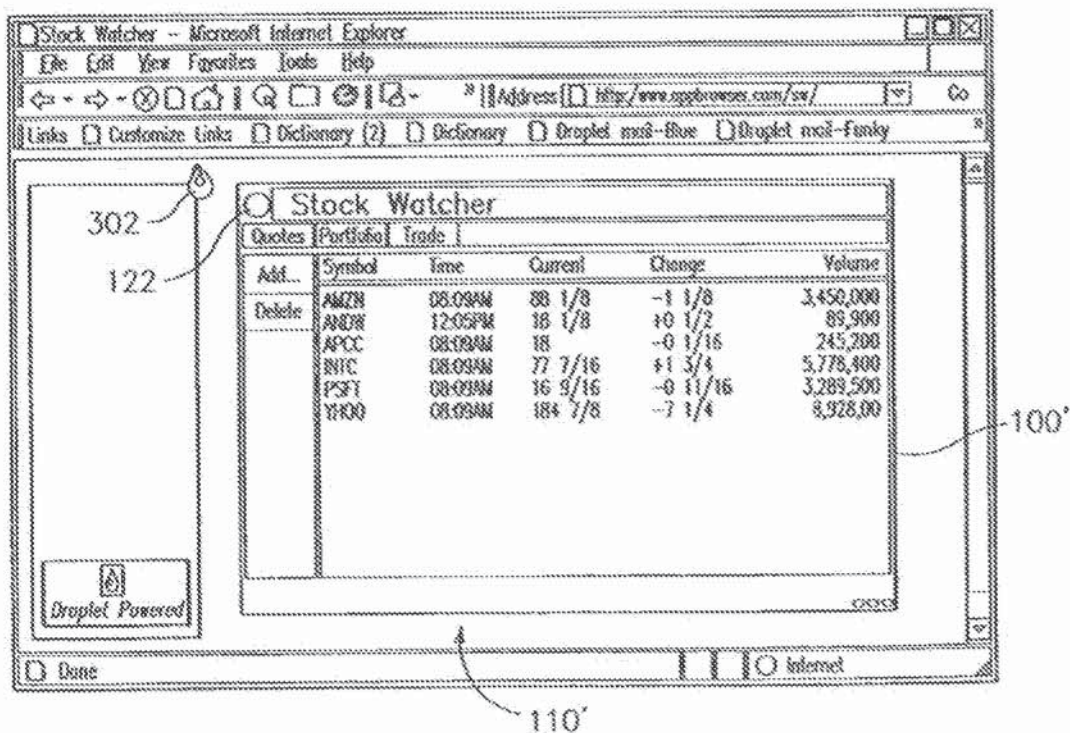


FIG. 4B

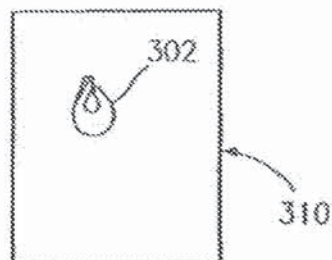


FIG. 4C

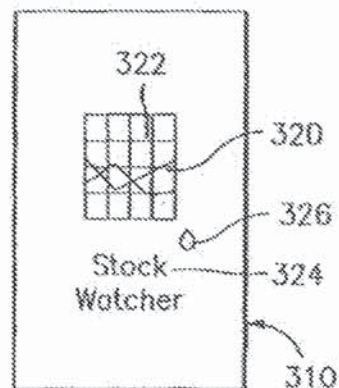


FIG. 4D

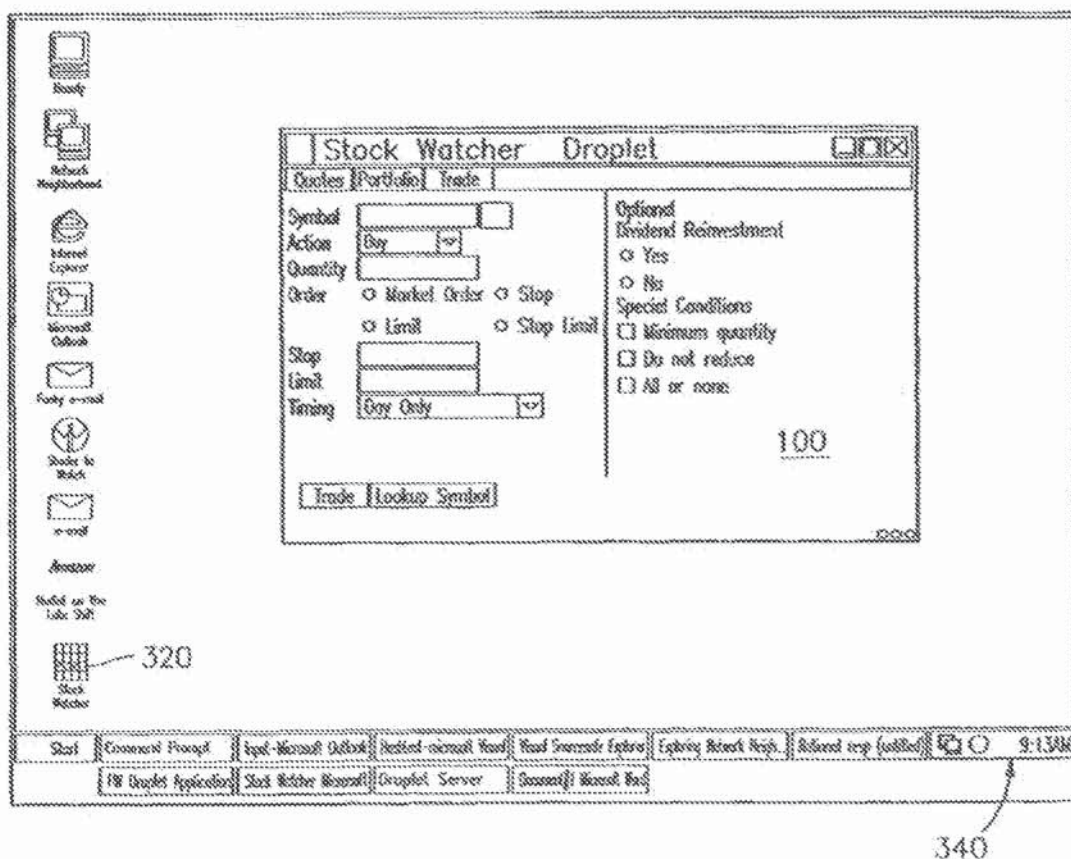


FIG. 5

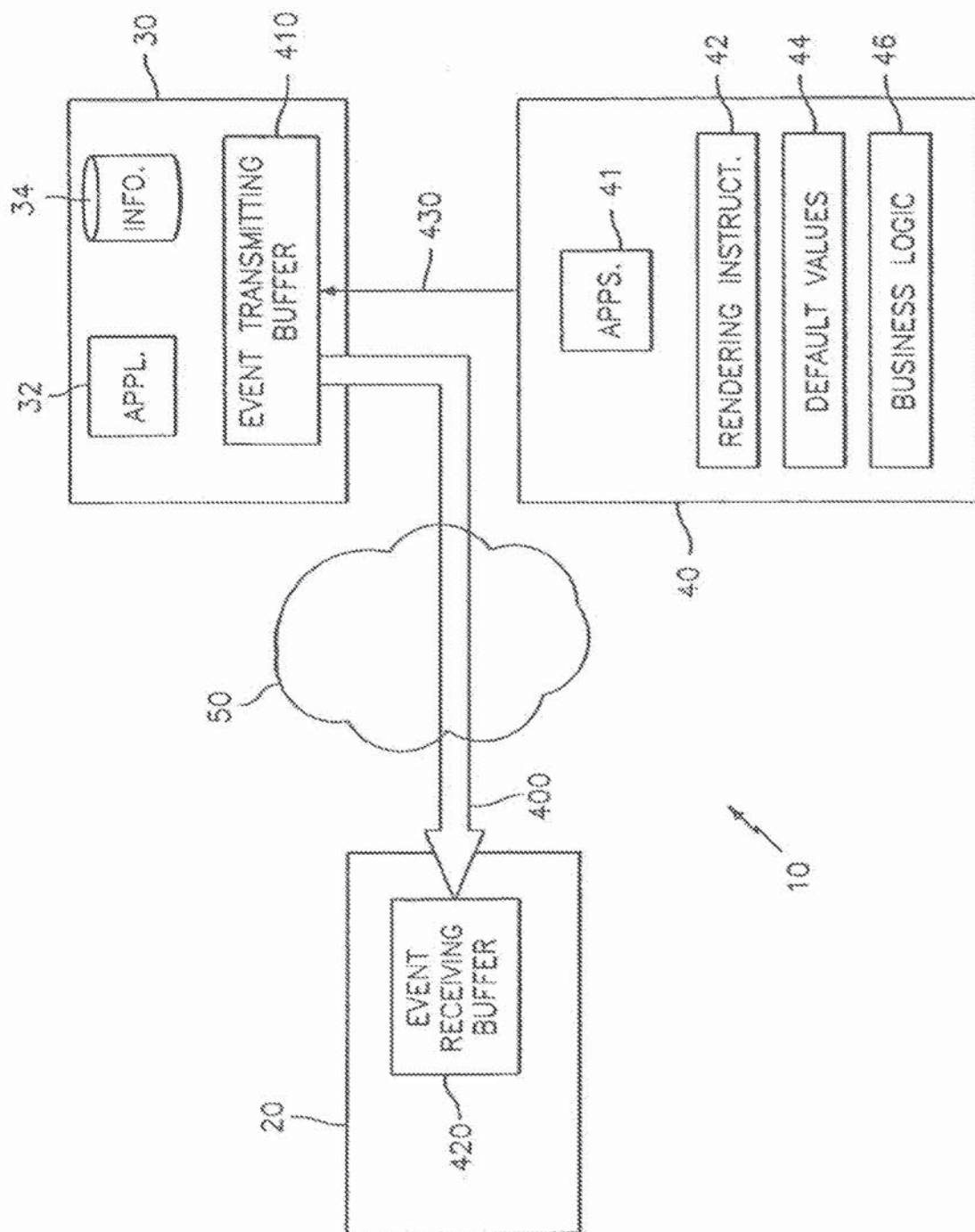


FIG. 6

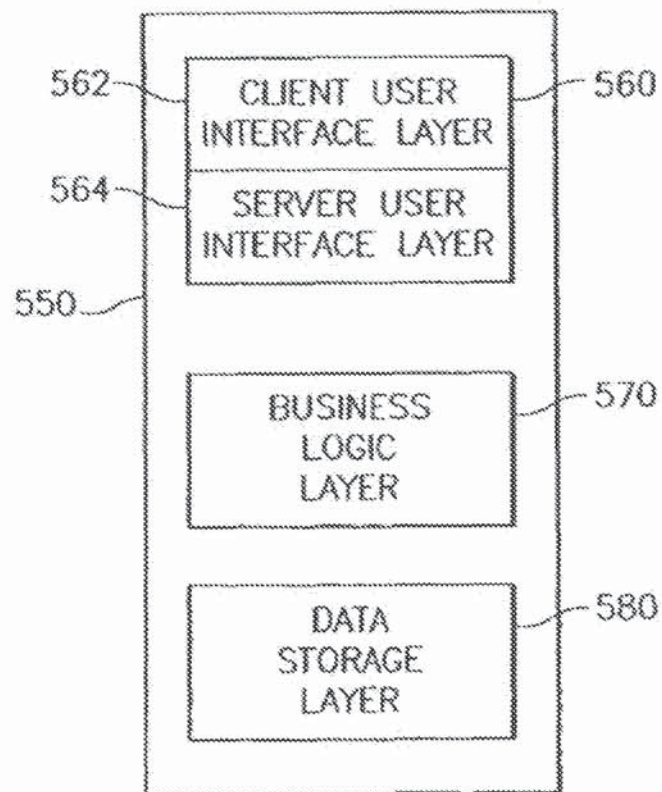
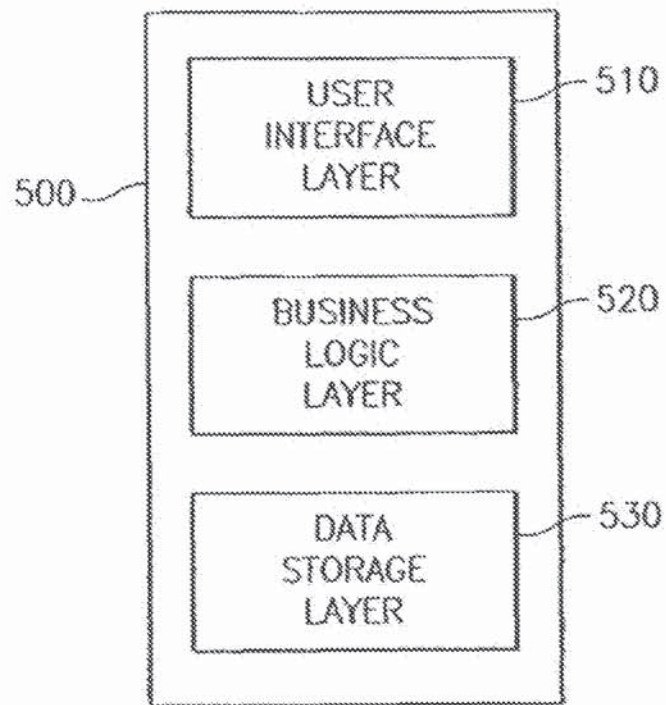
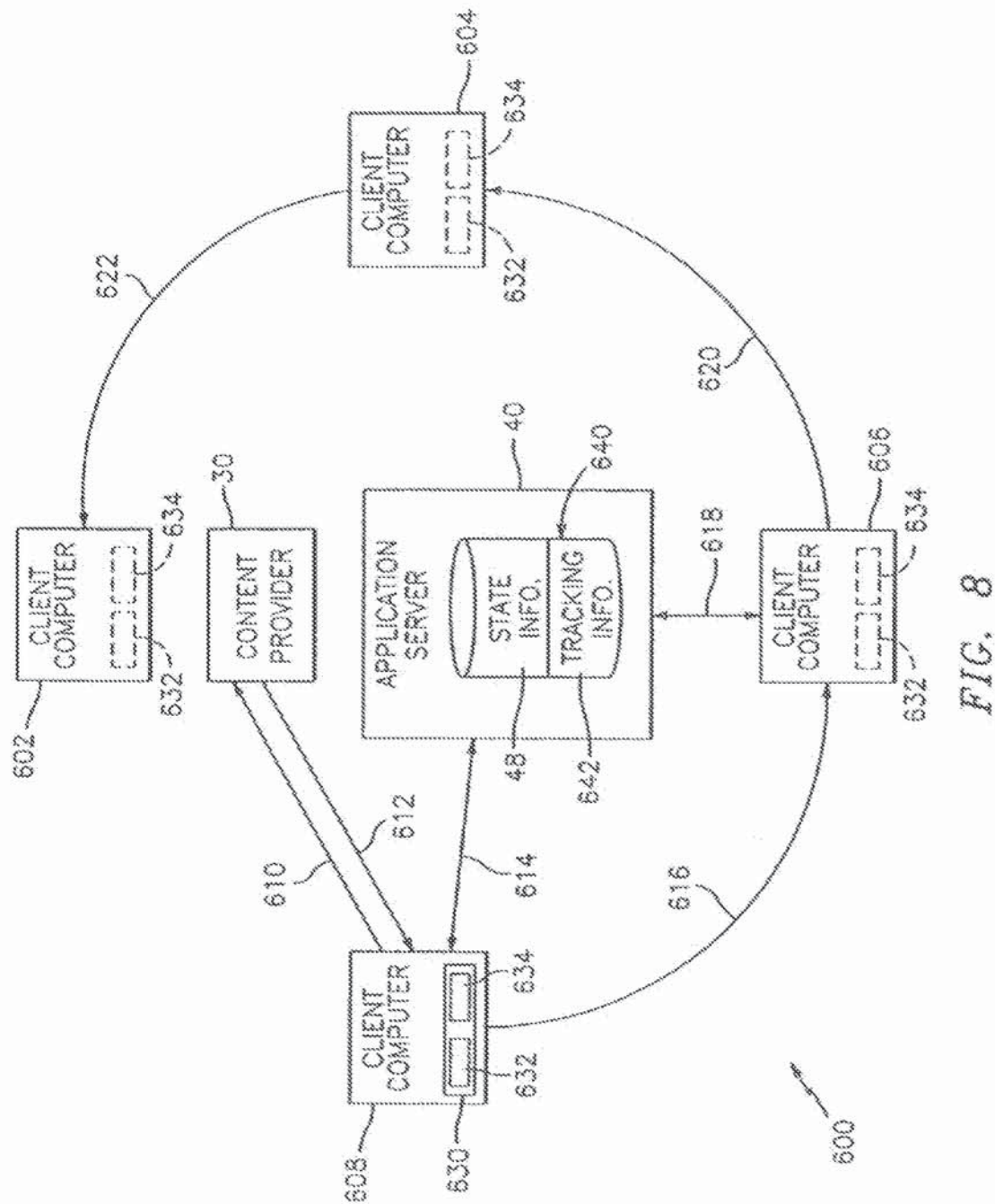


FIG. 7



US 7,502,838 B2

1

SYSTEM AND METHOD FOR DELIVERING REMOTELY STORED APPLICATIONS AND INFORMATION

The present application is a Continuation application of U.S. application Ser. No. 09/599,382 filed Jun. 22, 2000, which has issued as U.S. Pat. No. 6,687,745 on Feb. 3, 2004.

CROSS REFERENCE TO RELATED DOCUMENTS

Priority is herewith claimed under 35 U.S.C. §119(e) from copending Provisional Patent Application No. 60/153,917, filed Sep. 14, 1999, entitled "METHOD AND SYSTEM FOR DELIVERING APPLICATIONS IN CLIENT/SERVER ENVIRONMENT" by Louis M. Franco et al. The disclosure of this Provisional Patent Application is incorporated by reference herein in its entirety.

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

The present invention relates generally to computer processing systems and, more particularly, to an object-oriented approach for delivering interactive links to applications and information stored in remote sources of a network.

BACKGROUND OF THE INVENTION

An important aspect of any computing system is the interface between a human user and a computer. An early type of interface was text based. In a text based interface a user communicates with the computer by typing text characters with a keyboard and the computer communicates with the user by displaying text characters on a display screen or monitor. Recently, graphical user interfaces (GUIs) have become increasingly popular. In a GUI, a computer communicates with a user by displaying graphics, including text and images, on a monitor and the user communicates with the computer by typing in textual commands and/or by manipulating the displayed images with a pointing device such as, for example, a mouse, track ball, pressure-sensitive pad, electronic stylus or pen, or like devices.

Many GUIs are implemented in a operating environment referred to as a window environment. In the window environment, the display on the monitor of the computer is represented as an electronic "desktop," each application program available on the computer may be represented as an "icon" on the desktop, and each application program running on the computer may be represented as one or more electronic "sheets" displayed in regions of the monitor referred to as "windows."

Generally speaking, an application program presents information to the user through a window by drawing or "rendering" images, graphics or text within the window region. The user communicates with the application program by selecting objects in the window region with a cursor, which is controlled by a pointing device, and/or also by typing informa-

2

tion into the keyboard. In the window environment, the objects and the window regions may be manipulated or moved about the desktop, and changed in size and appearance so that the user can arrange and/or customize the appearance of the desktop in a convenient manner. The manipulation of an object by using a pointing device to capture and move the object somewhere else on the desktop, from one window region to another window region or within a window region, is generally referred to as a "drag and drop" operation.

The window environment described above is generally part of the computer's operating system. The operating system also typically includes utility programs that enable the computer system to perform basic tasks, such as rendering objects on the display monitor, storing and retrieving information on a memory device, performing file operations and communicating with a network. The application programs interact with the operating system to provide a direct interface with the user and higher level functionality such as, for example, to perform word processing or other specific tasks. Accordingly, the computer includes mechanisms for coordinating the operation of the operating system and application programs, including the arrangement of objects on the display monitor.

Computer processing systems are often implemented in network configurations such as "client/server" configurations. In client/server configurations, a number of personal computers, work stations, portable and/or handheld devices or the like ("clients") are linked and communicate with one or more host computers ("servers"). The servers process requests from the clients for information and/or application programs shared by the clients over the network. Today, many local and wide area public and private networks are interconnected to form a world-wide network of networks generally referred to as the Internet. The Internet permits access to a wealth of information and services stored at individual nodes or sites on the network. A particularly important subset of the Internet is the World Wide Web or "web." The web introduces graphics capabilities to Internet communications.

Sites on the web, generally referred to as web sites, are connected or linked together using a special communication protocol such as, for example, Hypertext Transport Protocol (HTTP), and a Uniform Resource Locator (URL) that includes a specific syntax for defining a network connection on the web. HTTP permits an exchange of information between users and the web sites that contains subject matter of interest to the users. From a user's perspective, the web appears as a collection of viewable documents, or web pages, residing at the various web sites. A user accesses the web pages by executing software, generally referred to as a web browser. A web browser running on the user's computer sends instructions (including the desired URL) for connecting to a specified host computer (i.e., web server) that stores web pages of interest to the user. A link, such as a hyperlink, is created under the communication protocol. By selecting links and employing a web browser, a user may "navigate" from one document to another, and from one web site to another, to access informational content and services available across the web.

One of the fastest growing uses of the Internet is as a vehicle of commerce. The world-wide access achieved by use of the web makes the web ideal for both small and large businesses who may enter the market by "launching" web sites offering their goods and/or services for sale. As a result, the informational content and services available to users of the web are considerable and are growing more so, everyday.

Despite the success of the web, problems persist for both users and businesses. For example, users need to locate items or services of interest from the vast number of sites on the web

US 7,502,838 B2

3

and, similarly, businesses need to distinguish themselves from competition to attract and hold the interests of desirable customers.

Search engines assist users in locating content and services of interest. However, even armed with searching tools, locating and retrieving informational content and services of interest from the many hundreds of thousands of web sites currently operating over the web may be cumbersome. The results of searches typically yield so many web sites that the effort to review the results and narrow the search may be overly cumbersome.

To distinguish their goods and/or services on the web, businesses often advertise their web site names in conventional media such as television and/or print advertisements. Businesses may also employ a web based solution in which rich, visually pleasing advertisements are incorporated in a number of popular web sites. The advertisements, generally referred to as "banner ads," often include links to the business' web site. Therefore, when the banner ad is selected, the user is transferred to the business web site where further information regarding the goods/services are displayed.

A perceived disadvantage of existing systems and methods for locating sites of interest and/or for attracting customers is that the links to the sites of interest generally must be acted upon from the page they are displayed or they are lost. In other words, conventional systems and methods present web sites that may include static links to other web sites. If a user is interested in viewing the other web site then the user must navigate to it or record the URL of the site for later use. As can be appreciated, it may be undesirable for the user to leave the current site to follow the link to the other site as the user may be performing an important and/or time sensitive task. As a result, a potentially valuable marketing or sales opportunity may be lost.

Facilities presently exist for storing an address (URL) of a web site currently being displayed. One such facility is referred to as a "bookmark." Once created, bookmarks offer a means of retrieving the URL of a particular web site and directing the user's browser to display the page residing at the URL. Bookmarks eliminate the need for the user to manually enter the URL of a site of interest or to retrace (re-navigate) a path through the Internet to arrive at the web site through a known link. However, bookmarks are limited in two respects. Firstly, a web page must be displayed before the URL corresponding to the web page can be stored as a bookmark. Secondly, bookmarks do not maintain information pertaining to a previous operating state of the web site. For example, a bookmark may return a user to a previously displayed web page, such as a form for completing a commercial transaction, but information that may have been completed on the form is generally not saved. That is, the completed information is generally not stored unless the information is made available through another tracking facility referred to as a "cookie." Cookies maintain tracking information on the user's computer that may be referenced once the browser reloads the desired web page and invokes the application included therein. Once the application is invoked, information that was previously entered and stored in the cookie may be restored in the application. Cookies, however, are generally time-sensitive and may expire before a user attempts to re-navigate to the site of interest. Also, cookies are only stored on the computer where the original transaction occurred. If the user accesses the site from another computer, the tracking information is not available.

Therefore, there is a need for storing an interactive link on a user's computer which, when selected, retrieves and presents applications and/or information stored at remote loca-

4

tions across the network. There is also a need for the interactive link to include facilities for restoring previous operating states of the application as the application is re-presented at a user's computer.

OBJECTS OF THE INVENTION

Accordingly, it is an object and advantage of this invention to provide interactive links to applications and information remotely stored across a network.

It is another object and advantage of this invention to provide graphical representations of interactive links to remotely stored applications and information, the graphical representations being downloadable to client computers and selectively employed to retrieve and present the remotely stored applications and information on the client computers.

It is still another object and advantage of this invention to provide graphical representations of interactive links to remotely stored applications and information, the graphical representations being downloadable to and transferable between one or more client computers for selectively retrieving and presenting remotely stored applications and information on each of the client computers.

It is yet another object and advantage of this invention to provide interactive links to remotely stored applications and information, wherein when selectively employed to retrieve and present the remotely stored applications and information on a client computer, a previous operating state of the applications and information may be restored.

It is still another object and advantage of this invention to provide graphical representations of interactive links to remotely stored applications and information, the graphical representations being downloadable to client computers and having a commercially recognizable appearance for identifying the remotely stored applications and information.

Further objects and advantages of this invention will become more apparent from a consideration of the drawings and ensuing description.

SUMMARY OF THE INVENTION

The foregoing objects and advantages are realized by methods and apparatus in accordance with embodiments of this invention, wherein interactive links for retrieving applications and information from remote sources in a network configured computer processing system are presented.

In one embodiment, a method for delivering the interactive links includes storing at a first remote source (e.g., a content provider) informational content (e.g., applications and information) having computer program code embedded therein. The method includes retrieving the informational content by a client computer of the system over a first communication connection to the content provider. In response to the retrieval of the informational content, the method further includes executing at the client computer the computer program code embedded in the informational content for establishing a second communication connection to a second remote source (e.g., an application server) on the network, and presenting via the second communication connection to the application server remotely stored applications and information. The remotely stored applications are executable at the application server for providing functionality of interest to the client computers. The information presented over the second communication connection includes at least presentational information providing at least one of instructions for rendering components of the remotely stored application, default

US 7,502,838 B2

5

parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.

The method also includes storing on the client computer an interactive link for selectively re-establishing the second communication connection to the application server for invoking and presenting the remotely stored application and information on an as-needed basis.

Additionally, the method includes downloading to the client computer a graphical representation of the interactive link from within the informational content delivered over the first communication connection, associating to the graphical representation a file containing information representing an operating environment of the client computer and a network address of the application server, and displaying the graphical representation on the client computer. Preferably, the downloading of the graphical representation is achieved in an improved drag and drop operation.

In accordance with a preferred embodiment, the graphical representations of the interactive links includes information for identifying the respective remotely stored application and information invoked by the interactive link, and a symbol indicating that the graphical representation is an interactive link. In one embodiment, the identifying information includes at least one, or both, of an icon and a label representing a commercially recognizable brand name of the remotely stored application and information, and the symbol is a tear-drop-shaped symbol.

Once the interactive link and the graphical representation are downloaded, the graphical representation may then be selected. In response to the selection, the method includes accessing the contents of the file associated to the graphical representation and re-establishing the second communication connection to the application server for invoking and presenting the remotely stored application and/or information associated to the interactive link.

In yet another embodiment, the method includes storing at the application server information representing a first operating state of the remotely stored application and/or information for a particular user, and in response to the selection of the graphical representation, presenting the first operating state information as the remotely stored application and/or information is presented to that particular user.

In a still further embodiment, the method includes transmitting a copy of the interactive link to a next client computer such that the next client computer is operable for selectively establishing the second communication connection to the application server for invoking and presenting, at the next client computer, the remotely stored application and information. In this embodiment, information may be stored for tracking transmissions of interactive links between the client computers of the network-configured computer processing system.

In a preferred embodiment, the network-configured computer processing system includes an Internet connection and the client computer includes a desktop-based repository for links to the remotely stored applications and information and an Internet-based repository for the links. Accordingly, the interactive links may be selectively stored in the desktop-based repository or in the Internet-based repository. Alternatively, the links are stored in both of the desktop-based repository and the Internet-based repository.

In accordance with the present invention, a computer processing system is configured and operates as follows. The system includes a plurality of client computers that each includes an application program code, an operating system program code and a presentational client program code oper-

6

ating thereon. The application program code requests informational content, applications and information stored remotely across a network. The system also includes a first server computer (e.g., the content provider) that stores informational content. Predetermined ones of the informational content include computer program code embedded therein. The content provider delivers the informational content in response to requests from the client computers. A second server computer (e.g., the application server) is included within the system. The application server stores applications that may be invoked for presenting functionality of interest at the client computers. The application server also stores information such as presentational information including at least one of instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.

The system also includes a communication connection between a requesting client computer and the application server. The communication connection is formed in response to an initial delivery of one of the predetermined informational content and an execution of the computer program code embedded therein. The presentational client program code, utilizing the communication connection, presents functionality of the remotely stored applications and information on the requesting client computer. The communication connection is disconnected upon termination of the remotely stored applications and information. The system further includes an interactive link that is stored at the requesting client computer for selectively reforming the communication connection to the application server, re-invoking the remotely stored applications and information, and re-presenting the functionality of the remotely stored applications at the requesting client computer.

Preferably, the predetermined ones of the informational content include the interactive link. When stored on the client computer, the interactive link includes a graphical representation of the remotely stored applications and information presented on the client computer and a file containing information representing an operating environment of the client computer and a network address of the application server computer. In this preferred embodiment, the graphical representation includes a symbol for indicating that the graphical representation is of an interactive link and at least one of an icon and a label representing a commercially recognizable brand name of the remotely stored applications and information.

In another embodiment, the application server includes a data repository for storing information representing a first operating state of the remotely stored applications and information presented to a particular user at a client computer. Accordingly, the system further includes a device for retrieving, delivering and presenting the previous operating state information from the data repository as the remotely stored applications and information are presented to the particular user at the same or another client computer.

In yet another embodiment, the system includes a device for transmitting and storing a copy of the interactive links at a next client computer. When the links are stored, the next client computer is operable for selectively requesting the remotely stored applications and information and forming the communication connection between the next client computer and the application server. The system further includes a data repository

US 7,502,838 B2

7

tory for storing information for tracking transmissions of interactive links between the client computers.

BRIEF DESCRIPTION OF THE DRAWINGS

The above set forth and other features of the invention are made more apparent in the ensuing Detailed Description of the Preferred Embodiments when read in conjunction with the attached Drawings, wherein:

FIG. 1 is a block diagram of an exemplary droplet-enabled client/server system constructed and operating in accordance with one embodiment of the present invention;

FIG. 2 illustrates a web page including an exemplary droplet-enabled application in accordance with one embodiment of the present invention;

FIG. 3 is a simplified flow diagram illustrating an exemplary flow of messages between an exemplary droplet-enabled client computer and an application server operating in accordance with one embodiment of the present invention;

FIGS. 4A-4D illustrate an improved drag and drop operation in accordance with one embodiment of the present invention;

FIG. 5 illustrates a selection of an exemplary, locally stored interactive link and the resulting invocation of a corresponding, exemplary droplet-enabled application;

FIG. 6 is a simplified block diagram of the system of FIG. 1 wherein an event channel communication scheme is particularly highlighted;

FIG. 7 illustrates a logical structure of an exemplary droplet-enabled application as compared to conventional application structure; and

FIG. 8 is a simplified block diagram of the exemplary droplet-enabled client/server system constructed and operating in accordance with one embodiment of the present invention.

Identically labeled elements appearing in different ones of the above-described figures are intended to refer to the same elements but may not be referenced in the description for all figures.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 illustrates the general architecture of a client/server configured computer processing system 10 constructed and operating in accordance with the present invention. For illustration purposes, FIG. 1 shows a single client computer 20 operatively coupled over a network 50 to a content provider 30 and an application server 40. However, the present invention contemplates (as discussed below with reference to FIG. 8) that the system 10 includes a plurality of client computers operatively coupled over the network 50 to two or more host/server computers. While the content provider 30 and the application server 40 are referred to herein as two separate computers, it is within the scope of the present invention for these computers 30 and 40 to be the same physical computer serving two logically separate functions. In a preferred embodiment, the network 50 is a packet-switched network such as, for example, the Internet, an intranet or an extranet. It should be appreciated that it is within the scope of the present invention for the network 50 to include wired and wireless connections between the client computers and server computers.

The content provider 30 services requests from the client computer 20 for informational content 36 such as, for example, applications 32 or information 34 shared by users over the network 50. In accordance with the present inven-

8

tion, the informational content 36 delivered to the client computer 20 by the content provider 30, over a communication connection 52, may include one or more links 62 and droplets 64. For example, a document 60 delivered to the client computer 20 includes a link 68 and a droplet 70. In some embodiments, a droplet™ is an application, or set of applications, that executes on the client computer 20 and, in cooperation with a presentation client 25 (installed at the client computer 20) and an operating system of the client computer 20, shown generally at 80, establishes a communication connection 54 with the application server 40 for retrieving information 43 and presenting functionality of remotely stored applications 41 on the client computer 20. The informational content 36 containing droplets™ may hereinafter be referred to as "droplet-enabled" content.

In one embodiment, the requested applications 32 and/or information 34 include functionality presented on the client computer 20 by the applications 41 executing remotely on the application server 40. That is, the application 41 is invoked by the droplet 70, as the droplet 70 executes on the client computer 20. As discussed in detail below, the links within the delivered informational content 36 (e.g., link 68) may be stored locally on the client computer 20 as interactive links 72 that, when selected, retrieve the information 43 and invoke the applications 41 to present the functionality of the applications 41 in the client computer. In particular, when performing a subsequent retrieval of the functionality presented by the applications 41, the applications 32 and/or information 34 that originally provided the link 68 (now locally stored as the interactive link 72) to the applications 41 need not be retrieved. Rather, the interactive link 72 can be employed to directly invoke and execute the applications 41 on the application server 40 to provide the requested functionality at the client computer 20. The initial and subsequent retrieval of droplet-enabled content and applications is discussed in further detail below.

Initially, droplets™ 64 execute as the informational content 36 containing the links 62 and droplets™ 64, e.g., the document 60 containing the link 68 and the droplet 70, is delivered to the client computer 20. That is, the droplet 70 executes and notifies the droplet presentation client 25 that it has been loaded to the client computer 20. Next, the droplet 70 cooperates with the droplet presentation client 25 and the operating system software 80 of the client computer 20 to establish the communication connection 54 to the application server 40. The link 68 is an interactive link presented within the informational content 36 for invoking the applications 41 and retrieving information 43 residing at remote locations on the network 50. Once the communication connection 54 is established, the droplet 70 transmits a startup message to the application server 40. In response to the receipt of a startup message from the droplet 70, the application server 40 executes the associated application 41 to present the application functionality and information 43 to the client computer 20.

In accordance with the present invention, droplets™ (e.g., the droplets 64 and 70) are dynamic and "thin" applications. That is, the droplets™ generally include information identifying the operating environment on the client computer 20, the application server 40 to connect with and an application on the server 40 that is run to deliver the requested functionality to the client computer 20 once the connection is made. The information identifying the operating environment on the client computers 20 provides information to the application server 40 regarding the operating system and hardware capabilities of the particular client computer 20 that requested the droplet-enabled content 36. That is, the plurality of client

US 7,502,838 B2

9

computers 20 may include computer workstations, personal computers and portable devices such as, for example, laptop and notebook computers, PalmPilot[®] and internet-enabled radio telephones. As is apparent to those in the art, each such device platform includes differing user interfaces. As such, not all client computers 20 are capable of presenting, for example, full color, high-resolution graphics. By providing the operating environment of the requesting client computer 20 to the application server 40, the application server 40 provides information 43 to present the requested applications 41 on the client computer 20. The information 43 includes, for example, instructions 42 for rendering graphical objects within the presented applications 41, default parameters or data values 44 displayed within the applications 41 and application-specific business logic 46 for processing inputs to the applications 41.

In accordance with one aspect of the present invention, a droplet application developer creates droplet-enabled applications or served versions of each application for presenting particular functionality to client computers having differing user interface ("UI") requirements. For example, a droplet-enabled email application may be implemented a number of ways such that a first version may operate on a personal computer having capabilities for providing full color, high-resolution graphics and a second version for operating on an internet-enabled radio telephone having only text-processing capabilities. In accordance with this aspect of the present invention, a droplet communicates one of the differing client environments and, in particular, client UI requirements, to the application server 40 which automatically provides, for example, the first version to a requesting personal computer and the second version to the requesting radio telephone. Alternatively, the droplet[™] could determine UI requirements from the client operating system or other locally stored data.

Since droplets[™] store a limited amount of information, that is, only the information necessary for establishing a connection (e.g., the communication connection 54) to an application server 40, droplets[™] require a minimum amount of storage space on a client computer, for example, no more than about 500 k bytes. Similarly, during execution, droplets[™] consume no more than about 2 M bytes of heap (dynamic memory allocation space). Therefore, a large number of droplets[™] may execute simultaneously without significantly reducing the memory available for other, concurrently executing applications. The droplet presentation client 25 requires local storage of about 1 M bytes for a complete installation. Once the droplet presentation client 25 is installed locally, the client computer 20 is "droplet-enabled."

It should be appreciated that, in one embodiment, the droplets (i.e., droplet 70) may include rendering instructions, parameters and/or data values that override the default information (i.e., the instructions 42 and values 44) received from the application server 40.

Preferably, the communication connection 54 initiated by the droplet 70 between the client computer 20 and the application server 40 is transparent to the user. That is, when the requested informational content 36 is delivered and presented on a display device of the client computer 20, the droplet 70 executes and, with the presentation client 25, establishes the communication connection 54. As noted above, the informational content 36 may include links to remotely stored applications 41, that are invoked to present functionality at the client computer 20. Presentational information 43 to present and serve, for example, various graphical objects (GUI objects) such as windows, edit boxes, drop-down lists, check boxes, buttons and/or labels included within the application

10

41, is received over the communication connection 54 as the functionality is presented on the display device or monitor of the client computer 20.

FIG. 2 illustrates a droplet-enabled application 100 entitled "Stock Watcher" that is presented on the desktop of the client computer 20. In this exemplary embodiment, the Stock Watcher application 100 is presented within a window on a web page, shown generally at 110, utilizing the MICROSOFT[®] Internet Explorer[™] web browser application. As should be appreciated, the present invention contemplates other presentational mechanism for droplet-enabled applications, that is, non-web based delivery such as a stand alone application program provided via a CD disk, floppy disk or like storage media. Other exemplary mechanisms are discussed in detail below.

As is known, web pages such as web page 110 are constructed using software such as the Hypertext Markup Language (HTML). A web page (i.e., web page 110) is droplet-enabled by embedding one or more droplets[™] within the page. Preferably, the droplets are embedded by using software commands such as, for example, "<applet>" or "<embed>" tags of the HTML language. As a web browser encounters a droplet-enabled web page, such as web page 110, instances of the droplets are invoked to initiate the communication connection 54 with the application server 40, as discussed above. Therefore, as a user navigates to web page 110, the droplets associated with the Stock Watcher application 100 are encountered in the web page 110 and execute. A droplet, in cooperation with the droplet presentation client 25, delivers the Stock Watcher application 100 over the communication connection 54, and presents the application 100 within web page 110. It should be appreciated that the user is generally unaware of the communication connection 54 established to the application server 40, as web page 110 and its corresponding GUI objects, e.g., text box 112, drop-down list 114, radio buttons 116 and command buttons 118, are presented within the web page 110 as the page 110 is loaded.

The Stock Watcher application 100 also includes an object referred to as a droplet handle 120. FIG. 2 presents the droplet handle 120 in a preferred configuration representing a tear-drop-shaped icon on the application's title bar. As is discussed in detail below, droplet handles may be icons, graphical images, or strings of text, that may be selected and downloaded to store, on a client computer 20, the interactive links (e.g., links 72 of FIG. 1) to droplet-enabled applications 41 remotely stored across the network 50.

In accordance with the present invention, droplets[™] are generic or universal applications that can be implemented on a variety of computer hardware and software platforms. For example, and as discussed briefly above, the droplet-enabled client/server system 10 may be implemented in an internet environment. In this respect, droplets[™] may include a downloadable Java applet, ActiveX controls, a browser plug-in written in the Java, C++ or other suitable programming language, or other multimedia objects, that are invoked as a web page is loaded. In the web based implementation, the droplets[™] cooperate with the droplet presentation client 25 and the web browser running on the client computer 20 to establish the communication connection 54 to the application server 40 and to present the droplet-enabled applications 41 and information 43 on the web page.

Alternatively, the droplets[™] may be included within a stand-alone software application that is installed and executes on the client computer 20. As connection to the application server 40 is still desirable in the stand-alone implementation, the droplets[™] include capabilities for establishing the communication connection 54 to the application server 40 and for

US 7,502,838 B2

11

presenting the droplet-enabled applications 41 and information 43 for use within the stand-alone application. Therefore, the droplets™, the droplet presentation client 25, the local operating system 80 of the client computer 20 and the web browser or stand-alone software program executing on the client computer 20 cooperate to establish the communication connection 54 to the application server 40 through conventional means such as, for example, with socket connections or by transmitting and receiving HTTP transactions.

As is apparent to those of skill in the art, the cooperation between the droplets™, the droplet presentation client 25, the local operating system 80 of the client computer 20 and the web browser or stand-alone software program executing on the client computer 20 to establish the connection to the application server includes the communication and sharing of information and a variety of functions and programming interfaces (generally referred to as APIs). That is, in the preferred embodiment of the present invention, standard operating system interfaces are utilized to access, create, control and manipulate features and functions of the client computer 20 such as, for example, window management, GUI and multimedia object management, system service procedures to establish socket and network connections, and generally any integration between the remotely executing applications, droplets, locally executing application and underlying operating system.

Whether in a web based or stand-alone implementation, the information 43, that is, instructions 42 for rendering graphical objects within the delivered droplet-enabled applications 41, default parameters or data values 44 displayed within the droplet-enabled applications 41 and application-specific business logic 46 for processing inputs to the droplet-enabled applications 41, is provided by the application server 40, in accordance with the operating environment of the requesting client (e.g., the client computer's operating system, user interface and hardware capabilities).

In one embodiment, the information 43 is provided by the application server 40 to the client computer 20 and stored therein in anticipation of a later use in which the user can not, or prefers not, to establish the communication connection 54, e.g., the internet connection, to the application server 40. In this embodiment, for example, the remotely stored application 41 and information 43 are downloaded and locally installed and/or stored at the client computer 20. The locally installed application 41 may be invoked to execute on the client computer 20 by selecting the link, as discussed above. In one aspect of the present invention, the application server 40 may evaluate the capabilities of the client computer 20 and automatically download all or a portion of the application 41 and/or information 43 for local installation on the client computer 20. In this respect, the locally installed application (in the form of, for example, native executables) may execute eliminating, during at least a portion of its execution, the need to receive rendering instructions 42, default parameters 44, or application-specific logic 46 from the application server 40.

In accordance with one aspect of the present invention, a network communication protocol is defined for transmitting information between the droplet-enabled applications 41 and the application server 40. The protocol includes a number of message formats wherein properties of and events pertaining to components, such as the GUI components, of actively operating droplet-enabled applications 41 are communicated between the client computer 20 and the application server 40. The message formats include, for example:

1. Event Notifications—messages transmitted from a client computer 20 to the application server 40 reporting that a value or attribute of a component of the GUI has

12

been altered. Events include, for example, data entry into text boxes and drop-down lists of the GUI, selection ("clicking") of GUI components such as radio and command buttons. Messages are also transmitted in response to other pointing device or keyboard driven actions such as, for example, drag and drop events as an e-mail message is moved to a folder within a droplet-enabled e-mail application.

2. Update Commands—messages transmitted from application drivers, e.g., the application specific logic 46 supporting a droplet-enabled application, on the application server 40 to the client computer 20 requesting action within screen components of the system 10, such as GUI components within the delivered content.
3. Window Commands—messages transmitted from the application drivers on the application server 40 to the client computer 20 that specify that a generic window should be opened or closed, or that the droplet-enabled application should present one of a set of standard dialogs (e.g., a message box, a question box or a menu). The droplet-enabled application may transmit a window command to notify the application server 40 that a user has manually closed an active window on the desktop.
4. Browser Commands—messages directed to a web browser running on the client computer 20 in which the droplet-enabled application is included. If the droplet-enabled application is included within a stand-alone software application that does not include a web browser, then these messages are sent to a default browser.
5. Session Commands—messages transmitted from either the client computer 20 or the application server 40 for initiating, suspending, or terminating sessions between the client computer 20 and the application server 40.
6. Service Directory—messages sent by the application server 40 to inform the client computer 20 of services that are available at the application server 40. Enabled client computers 20 that receive service directory messages can present the directory of services to users and can transmit a user's request for a particular service back to the server for processing.

The above described communication protocol and message formats provide a real-time push/pull messaging scheme between the user interface, and GUI components included therein, of an active droplet-enabled application presented on client computer and application drivers supporting the user interface from the application server 40. For example, an executing droplet-enabled application such as the Stock Watcher application 100 receives real-time information such as revised stock prices or shares available for trading as they are posted by an issuing agency. The user need not manually request a refresh of GUI components presenting the pricing and availability information, rather the client computer automatically receives the real-time updates to the components as the droplet-enabled application runs on the application server 40.

In one embodiment, a continuous open socket communication connection is maintained between the client 20 and server 40 to receive the real-time updates as they are posted (push transaction). In another embodiment wherein, for example, a firewall or proxy server security platform does not permit the continuous open socket connections, a request for updates is periodically transmitted by the client computer 20 to the application server 40. The automatic pseudo-real-time messaging strategy embodies a pull type messaging mechanism. It should be appreciated that the system 10 seamlessly implements this update messaging mechanism by first

US 7,502,838 B2

13

attempting the real-time push of update information, and if the application server 40 is not permitted to maintain an open connection communication channel, the server 40 informs the client computer 20 to employ the pull-type messaging mechanism by periodically requesting update information. In each of the above-described messaging schemes, the user at the client computer 20 receives the update information without manually requesting a refresh of the information.

FIG. 3 is a diagram illustrating a typical sequence of messages transmitted between the client computer 20 and the application server 40. As is shown, connect/startup messages 200 are transmitted to the application server 40 by the droplet 70 running on the client computer 20 when the droplet-enabled content 36 is delivered to the client computer 20. The startup messages 200 include information identifying the operating environment on the client computer 20, the application server 40 to connect with and an application on the server 40 that is run to deliver content to the client computer 20 once the connection is made.

As discussed above, the information identifying a client computer's operating environment provides the application server 40 a means for presenting a droplet-enabled application having a user interface customized to the capabilities of the client computer's operating systems and hardware.

The application server 40 responds to the startup messages 200 by transmitting messages 210 including information 43 (e.g., rendering instructions 42, default parameters and values 44, etc.) to present the droplet-enabled applications 41 on the client computer 20. In the web-based implementation, the droplet-enabled application 41 is preferably executing on the application server 40 as the messages 210 are provided to the client computer 20. During the course of executing the droplet-enabled applications 41 and presenting the functionality thereof on the client computer 20, messages 220 including event and update information are repeatedly communicated between the client computer 20 and the application server 40. As such, the droplets facilitate an interactive communication environment between the client computer 20 and the application server 40.

Finally, a message 230 including a terminate session command is transmitted to the application server 40 when, in the web-based implementation, the web page containing the droplet-enabled application is closed or the user navigates to another page, and, in the stand-alone implementation, as the stand-alone droplet-enabled application is closed or otherwise terminated.

The communication protocol and message formats (discussed in detail above) provide real-time communication between the client computers 20 and the application server 40. It should be appreciated that the communication protocol and messaging format transmit information that define differences between a current application state and a next application state, the information being referred to herein as net change information. Transmitting net change information permits, for example, a change of state of individual components on a screen as opposed to transmitting all screen data and repainting the entire screen from the transmitted screen data. By transmitting only net change information, communication resources such as, for example, bandwidth and transmission times are optimized. As such, droplet-enabled applications deliver rich application functionality over minimum network bandwidth and may operate over thin dial-up or wireless networks as well as broadband networks. Further, droplet-enabled applications supported from remote network sites appear as responsive as applications loaded and executing locally on the client computer 20.

14

The above described message formats and general sequence of messaging are utilized by the droplet-enabled applications 41 and the application server 40 to support, in cooperation with the presentation client 25 and the local operating system (e.g., operating system software 80), the following features and functions on the desktop of the client computer 20:

1. downloading interactive links to droplet-enabled applications and information;
2. branding of droplets;
3. transferring of links to droplet-enabled application/information between client machines;
4. securing an instance of a droplet-enabled application/information;
5. tracking droplet dispersion information;
6. providing an ubiquitous event channel;
7. persistent state maintenance;
8. mirroring desktop to webtop;
9. multiple platform implementation of droplets; and
10. security features.

Downloading Interactive Links to Droplet-enabled Application and Information:

As discussed in the Background Section of this Specification, objects displayed on a GCU (e.g., on the desktop or a window region thereof) may be captured and moved about the GUI in a "drag and drop" operation. In accordance with the present invention, the links to droplet-enabled applications within informational content 36 delivered to a client computer (e.g., a link to a droplet-enabled application presented within a banner ad on a web page) may be downloaded to the client computer via a drag and drop operation. That is, a droplet handle object incorporated on the banner ad of the delivered informational content 36 may be captured by, for example, selecting the droplet handle object, and moving (dragging) the handle about the client computer's GUI. When in a desired location or position on the client computer, the droplet handle object may be locally loaded (dropped) in an improved drag and drop type of operation.

Specifically, the present invention supports the downloading (e.g., dragging and dropping) of links to droplet-enabled applications within delivered informational content 36 from: (1) a web page onto the desktop; (2) a first window region onto another window region that accepts files; and (3) a web page or window region onto a directory or an application program's menu such as, for example, the Start Menu of the Microsoft Windows™ operating system software. Accordingly, locally stored links are selected to invoke and present, on the client computer 20, functionality provided by droplet-enabled applications, executing on the application server 40, and information on an "as-needed" basis.

With reference to FIGS. 4A-4D, a preferred embodiment of the downloading operation is disclosed. As shown in FIG. 4A, a droplet-enabled Stock Watcher application 100' includes a selectable droplet handle object, represented as a teardrop-shaped icon 120', located on its title bar. A pointing device (not shown) positions a cursor 300, represented as a hand having a pointing finger, above the icon 120'. The icon 120' is selected by, for example, holding down on a button associated with the pointing device (e.g., a left mouse button). The selected icon 120' is then moved (e.g., dragged) from a window 110' presenting the Stock Watcher application 100' to another location on the GUI by moving the cursor 300 from the window 110' to, for example, a portion 310 of the desktop of the client computer 20 (FIGS. 4B and 4C). The icon 120' is then released by disengaging the pointing device to place

US 7,502,838 B2

15

(drop) a graphical representation 320 of the Stock Watcher application 100 on the portion 310 of the desktop (FIG. 4D).

While the downloading of interactive links to droplet-enabled application and information is described above with references to FIGS. 4A-4D as preferably including a drag and drop operation, it should be appreciated that the present invention contemplates other means for downloading the links to a droplet-enabled application. For example, a link to a droplet-enabled application may be downloaded to a client computer by invoking a menu command to download a graphical object representing the link or by pressing a combination of keyboard keys at the client computer 20 such as "CTRL," "ALT" and "D."

In one embodiment, the cursor 300 is utilized to initiate a download to the client computer 20 of one of the interactive links in one of two ways. Firstly, and as discussed above with reference to FIGS. 4A-4D, the cursor 300 is positioned above an object representing the link, i.e., the icon 120', by manipulating the pointing device. As the icon 120' is selected to initiate the download, a visual characteristic of the cursor 300 is modified. That is, the hand having the pointing finger (FIG. 4A) is transformed to a cursor 302 where a closed hand encloses the teardrop shaped droplet symbol (FIG. 4B). To further illustrate the capture of the icon 120', the icon 120' is also transformed into a symbol 122 of, for example, an empty circle. As noted above, the cursor 302 is moved about the desktop by holding down on a button of the pointing device and manipulating the cursor 302. Once in a desired location on the client computer 20, e.g., on the desktop (FIG. 4C), the link is loaded to the client computer by releasing the button of the pointing device. Once released, by disengaging the mouse button, the link is loaded to the client computer 20 (FIG. 4D). It should be appreciated that the visual representations discussed above of the cursor 302 and icon 122 are merely illustrative and other visual representations of the selected cursor and icon are contemplated such that when selected, for example, an object may appear in reverse video, a frame may be drawn around the object or "handles" may appear at various corners of the selected object.

Secondly, the icon 120' may be selected by positioning the cursor over the icon 120' using the pointing device, depressing and releasing the button on the pointing device. As is generally known to those of skill in the art, performing a depressing and releasing operation on an object (referred to as "clicking" the object) is a separate operation from depressing, holding and moving an object. In response to the clicking operation, a submenu (generally referred to as a "pop-up" menu) appears at the current cursor location. The pop-up menu includes a list of options for downloading the link to, for example, the desktop or a window on the desktop. It should be appreciated that the list of options for downloading may appear in various manners. That is, each option of the list displayed in the pop-up menu may be represented by text describing the operation (e.g., "download to the desktop", "download to the start menu", etc.), a text label followed by a control such as a radio button or a check box, or a graphic icon. Each of the items is selectable for invoking a specific download operation.

In accordance with one aspect of the present invention, the graphical representation 320 of a droplet-enabled application, i.e., the Stock Watcher application 100, includes information for identifying the application and a symbol that indicates that the application is droplet-enabled. In FIG. 4D, for example, the graphical representation 320 is illustrated as a chart icon 322 and a label 324 entitled "Stock Watcher." The representation 320 also includes a "tear-drop" symbol 326 for representing that, when selected, the representation 320

16

invokes a droplet-enabled application, i.e., the Stock Watcher application 100. Thus, the graphical representation 320 is the link to the remotely stored Stock Watcher application 100. Preferably, the graphical representation 320 and, in particular, the application identification portion thereof, includes branding information. The branding of graphical representations of droplet-enabled applications and informational content is discussed in further detail below.

While the Stock Watcher application 100 included a single droplet handle object, i.e., the teardrop-shaped icon 120, it should be appreciated that droplet-enabled applications and informational content may include more than one downloadable item (link) for invoking various functional and informational features of the applications stored remotely. The item (link) may be an icon, an image or a string of text that, when selected, invokes a droplet-enabled application or retrieves droplet-enabled information from a remote location on the network.

As should be appreciated by those of skill in the art, the downloading (e.g., improved drag and drop) operation includes steps of interacting with the operating system 80 of the client computer 20 (standard operating system API routines). The interaction permits tracking of the pointing device during a drag and requesting that a file be generated on the target of a drop. When dropped, the file (e.g., files 74 of FIG. 1) is associated to the interactive link (e.g., links 72 of FIG. 1). The file includes information for re-establishing the communication connection 54 to the application server 40 as the link is selected, as is discussed below.

In accordance with the present invention, and as depicted in FIG. 5, selection of the interactive link invokes the droplet-enabled application and/or information, e.g., the selection of the graphical representation 320 invokes the Stock Watcher application 100. That is, selecting the representation 320 (link) triggers the droplet supporting the Stock Watcher application 100 to re-execute on the client computer 20. The droplet, in cooperation with the droplet presentation client 25 and the local operating system 80, re-establishes the communication connection 54 to the application server 40 for invoking and presenting functionality of the droplet-enabled Stock Watcher application 100.

The file generated during the downloading operation is accessed to support the re-presentation operation. As illustrated in FIG. 1, each of the files 74 are associated to a respective link 72 as a result of the downloading operation, e.g., link 1 to file 1 and link 2 to file 2, etc. The files 74 include information identifying the operating environment on the client computer 20, and the identities of the application server 40 to connect with and the server application that is run to present the Stock Watcher application 100 (or other content) to the client computer 20, once the connection is made. Preferably, the details/content of the file are determined by the parameters that embedded the downloadable item (link) into the informational content 36 delivered to the client computer 20. That is, for example, the commands used to add the teardrop-shaped icon 120 into the web page that included the Stock Watcher application 100 include a definition of the file to be created during any downloading operations to load the interactive link to the Stock Watcher application 100 on a local client computer.

In one embodiment, the file 74 includes a location of the application server 40, the application to be run on the server 40, the machine local addresses for the application server 40 and client computer 20, ports (buffers supporting inter-machine communication between processes), and droplet type (e.g., HTML, Java applet, a multimedia object, etc.). As discussed above, the file 74 may also include information to

US 7,502,838 B2

17

support presentation of the graphical representation 320 (e.g., rendering information, labels, window definitions). The presentational information included within the file 74 overrides the default information provided by the application server 40.

In another embodiment, the files 74 include a location (address) of the content provider 30 that originally delivered the droplet-enabled informational content. Under certain conditions, for example an error in establishing the communication connection 54, the droplet, the droplet presentation client 25 and the local operating system 80 may cooperate to re-establish the communication connection 52 and request redelivery of the informational content 36 that originally provided the link 72.

Accordingly, the present invention provides a mechanism for locally storing links to remote droplet-enabled applications and/or information. For example, FIG. 2 illustrates the web page 110 presenting the droplet-enabled Stock Watcher application 100. As described above with reference to FIGS. 4A-4D and 5, the droplet handle 120 (the link) to the Stock Watcher application 100 was downloaded from the web page 110 and locally stored as the graphical representation 320 on the portion 310 of the desktop of the client computer 20. As should be appreciated by those of skill in the art, the graphical representation 320 of the link is selected by, for example, double clicking the representation 320 with the pointing device. Once selected, the link causes the local operating system 80 to invoke the droplet supporting the droplet-enabled application. That is, the droplet cooperates with the droplet presentation client 25 and local operating system 80 to access the associated file and to re-establish the communication connection 54 to the application server 40 for re-delivering information for re-presenting the Stock Watcher application 100. Importantly, the Stock Watcher application 100 is, in effect, re-executed at the application server 40 to re-present to the client computer 20 the functionality of the Stock Watcher application 100 without having the application 100 locally loaded on the client computer 20 and also without re-navigating back to the web page 110 that originally presented the Stock Watcher application 100 to the user. By eliminating the need to locally store applications and/or to re-navigate to a web page containing a desired application, the present invention avoids two perceived disadvantages of conventional processes for invoking remote applications and retrieving remotely stored information.

Branding of Droplets:

In accordance with the present invention, the graphical representation 320 and, in particular, the application identification portion thereof, includes branding information. The branding information includes, for example, a name of the application or institution supporting the application, and a logo or other familiar and easily identifiable symbol of the application or institution. Generally speaking, institutions utilize branding information to uniquely identify a set of products and/or services of the institution in the hope of establishing in the minds of its customers an indication of certain levels of quality and trustworthiness. Branding is generally judged to be important in electronic commerce (e.g., Internet-based transactions) for branding enables customers to find, compare and select reliable businesses from the wide variety of competitive companies offering similar products and services on the web.

By branding graphical representations of droplet-enabled applications and informational content, the supporting or sponsoring institution may establish, for example, a commercially valuable presence on the Internet and, when downloaded, on the client computers. In accordance with the

18

present invention, not only is a locally stored interactive link provided for invoking remotely stored applications and information, but the link may also be visually customized to resemble a commercial image such as, for example a corporation's business name and/or logo.

In one embodiment, for example implemented under the MICROSOFT® Windows™ operating system, the desktop supports the ability to use custom code libraries to render icons for specific file types. The system 10 uses this capability to render application specific icons under Windows™. In Windows™, icon information may be stored in a file. When the desktop needs to draw the icon, it calls a library function that accesses the icon information in the files. The library function reads the file, builds the icon from the data therein, adds branding information (if any) and returns the image to the desktop for rendering. For droplet-enabled applications and/or information, therefore, files are maintained for all icons. The files include, for example, at least the droplet-enabled symbol (the teardrop symbol 326 of FIG. 4D) and, in one embodiment, the file is downloaded from the server 40 as a result of executing the downloading (e.g., improved drag and drop) operation discussed above, and the icon information is included within the file 74.

In another embodiment, for example implemented under the MACINTOSH® MacOS™ operating system, the client operating system natively supports unique icons for each file. On this platform, the icon is assigned when the file is first created, e.g., as a final step of the improved drag and drop operation. Under some circumstance, the proper icon may not initially be defined, e.g., if the file containing the rendering information was copied or dragged from an incompatible system. In those circumstance, when the client invokes the droplet-enabled application, the MacOS™ checks for the existence of a custom icon in the file. If the icon does not exist, it creates one from the icon information within the file.

In another aspect of the present invention, droplet-enabled applications may be "skinned." That is, a "look and feel" of a user interface for a particular droplet-enabled application includes aesthetically appealing or commercially branded visual and auditory characteristics that tend to distinguish the application from other applications running within the network. The visual and auditory characteristics may include, for example, distinctive colors, text fonts, control layouts, graphics or relative sizes and spacing of controls. A "skin" may reflect an advertising or promotional message, a favored item of interest such as, for example, a scene from a movie, video game or landscape. Additionally, the skin may vary characteristics of controls and labels to provide a multi-lingual presentation of a particular droplet-enabled application. That is, the skinning permits an application to be presented in varying languages such as, for example, English, Spanish, German, Italian, etc.

In accordance with the present invention, a skin may be chosen and applied in run-time without requiring a change in the droplet-enabled application's coding or business logic (i.e., the C++, Java, visual basic or other coding used to develop the applications need not be altered). Therefore, skins permit the user interface of a given droplet-enabled application to be customized (e.g., "private-labeled") from one implementation (e.g., website) to another. That is, the customized user interface may be configured as a commercially (branded) or aesthetically appealing interface giving each instance of the droplet-enabled application a desired look and feel. The customization and/or the ability to choose a skin may be provided under the direction of a droplet application developer, website webmaster, or end-user.

US 7,502,838 B2

19

In one embodiment, the skin that is applied as functionality of a droplet-enabled application is presented at the client computer 20 is selected from a library of available skins maintained at the content provider 30. A reference to the selected skin may then be downloaded and maintained within the link locally stored at the client computer 20. Alternatively, the skin may be predefined (e.g., hard coded to overrides user preferences) at the application server 40 so that as the droplet-enabled application is presented to the client computer 20 the predefined skin is included.

Transferring Droplets Between Users:

In accordance with one aspect of the present invention, droplets may be transferred between users and their respective client computers. For example, the droplet-enabled client/server system 10 includes the ability to transmit a message, e.g., to compose an email message, which contains a droplet™ and an associated link to a remotely stored droplet-enabled application and/or information. Preferably, the message is constructed to substantially ensure that the recipient is able to run the droplet™ and access the droplet-enabled applications or information, regardless of the platform of the receiving machine.

For example, a first user may download a droplet-enabled banner ad from a web site onto their desktop. The droplet-enabled banner ad may include, for example, an application for conducting an informational survey (e.g., a marketing survey) of importance to the sponsoring entity of the web site. In exchange for completing the survey, the sponsoring entity may offer a discounted price on some of its merchandise available for purchase on the web site. The first user considers the discounted price an attractive offer and, therefore, wishes to pass the offer onto associates. In accordance with the present invention, a message is created that includes the droplet-enabled banner ad, the droplet™ and the link to the remotely stored application and/or information supporting the survey (droplet-enabled survey application and/or information). The message is transmitted to associates of the first user. Depending on capabilities of the associates, or more precisely, on the client computers used by the associates, the droplet-enabled survey application and/or information may be immediately accessed and utilized by the associates.

In the present invention, the system 10 includes at least one of three possible strategies for transmitting and running applications on the associate's client computers, hereinafter referred to as a recipient computer. In a preferred embodiment, each of the three strategies is included within, for example, a message transmitting the link and, depending on the capabilities of the recipient computer, one of the three strategies may be employed to successfully receive the application.

Firstly, the transmitting message (e.g., email message) may include a droplet-enabled application and/or information in a file format as an attachment. If the recipient computer is droplet-enabled, i.e., operating in a similar way as client computer 20 wherein the droplet presentation client 25 is locally installed, then the attachment appears as a known file type and the recipient may open the droplet-enabled application and/or information file immediately. Once opened, the link to the droplet-enabled survey application may be downloaded (e.g., dragged and dropped) on the recipient computer. As a result, the file attachment is copied onto a desktop or other location on the recipient computer (e.g., the "start" menu). Once downloaded, the link (e.g., graphical representation 320 of the link) may be selected to initiate the communication connection 54 to the application server 40 so that the

20

droplet-enabled survey application and/or information may be executed and/or retrieved as discussed above.

In a second, exemplary implementation where the recipient computer is not droplet-enabled, the droplet-enabled survey application and/or information may be transmitted between parties by transmitting a message (e.g., an email message) containing a hypertext link to a web page. The web page includes a link for launching a web browser and for guiding the user through the process of droplet-enabling the recipient computer. That is, the web browser receives instructions for downloading the droplet presentation client 25 and links to the droplet-enabled survey applications and/or information.

In a third exemplary implementation, a message (e.g., email message) that transmits the droplet-enabled survey application and/or information includes a link to a web page that contains the droplet-enabled survey application and/or information as an embedded object thereof. This method is similar to the methods of retrieving droplet-enabled application and information discussed above with reference to the Stock Watcher application 100, with the exception that the recipient is directed specifically to the web site of interest without having to navigate to the site on their own. This implementation assumes that the web browser used by the recipient has the ability to host Java applets.

FIG. 8 is a simplified block diagram of a droplet-enabled client/server system 600 configured and operating in accordance with the present invention. In particular, FIG. 8 depicts two host computers (e.g., the content provider 30 and the application server 40) servicing requests for droplet-enabled applications and information from a plurality of client computers 602-608. The droplet-enabled applications and information are provided to a client computer either directly from the host computers 30 and 40 or droplets and links to the applications and information are transferred from another client computer. As the direct host-to-client delivery process has been described in detail above, reference is made generally to the process of transferring droplets and links by and between the client computers 602-608.

In FIG. 8, client computer 608 requests informational content from the content provider 30 (via a message 610). In response, the content provider 30 provides the requested informational content 630, which includes, e.g., a link 632 and a droplet™ 634 embedded therein (via message 612). Upon receipt, the droplet™ 634 executes on the client computer 608, and in cooperation with a droplet presentation client and the local operating system (not shown), establishes a communication connection 614 to the application server 40. The application server 40, in turn, presents (over the communication connection 614) to the client computer 608 the functionality of the droplet-enabled application, including information for presenting the application. The link 632 may be locally stored to selectively re-establish the communication connection 614 and re-present the droplet-enabled application. As described in detail above, this process represents the first method of presenting functionality of droplet-enabled applications and/or information to a client computer.

FIG. 8 also illustrates a second mechanism wherein droplets and links to droplet-enabled applications and/or information are transferred by and between the respective client computers 602-608. For example, the link 632 and droplet 634 downloaded to client computer 608 are transferred, via message 616, to the client computer 606. Once received and stored, the link 632 and droplet 634 (shown in dashed-lines to illustrate that the link 632 and droplet 634 were not downloaded directly to the client computer 606) may be selected to establish a communication connection 618 to the application server 40 whereby the droplet-enabled application and infor-

US 7,502,838 B2

21

mation for presenting the application may be passed to client computer 606. Similarly, the link 632 and droplet 634 may be passed from client computer 606 to client computer 604, via a message 620, and from client computer 604 to client computer 602 via a message 622. As should be appreciated, the passing of links and droplets facilitates the retrieval of droplet-enabled applications and/or information without requiring the user of a client computer to navigate to the content provider 30 that originally provided the link and droplet for the droplet-enabled application and/or information. In many respects, the second mechanism for presentation provides a more advantageous method of sharing network applications and information among a group of client computers than the first mechanism wherein each client computer with the group individually navigates to and downloads unique instances of links and droplets of droplet-enabled applications.

Securing Droplet-enabled Application and/or Information:

As generally discussed herein, a link to droplet-enabled applications and/or information is selected to invoke a unique instance of the application and/or information stored at a remote location on a network. In other words, each request for execution of a droplet-enabled application or retrieval of droplet-enabled information is satisfied by providing an individual copy (instance) of the requested application and/or information.

However, in accordance with one aspect of the present invention, a secured and individually identifiable instance of droplet-enabled applications and/or information may be shared and transferred between multiple client computers. That is, the present invention provides a keyed instance of a droplet-enabled application having a unique identification field referred to herein as a global unique identification field (GUID). The keyed instance may be transferred between one or more client computers such that information accessed and stored from within the keyed instance may be modified only by client computers that invoke the keyed (identification field specific) instance of the droplet-enabled application. The keyed instance is, in effect, secured such that only users knowing a value of the GUID field may specifically request and access the individual instance of the secured droplet-enabled application and/or information.

For example, the client/server system 10 includes a secured, or keyed, droplet-enabled photo-album application. A first user enters personal photos into a uniquely identifiable instance of the photo-album application, which has a GUID field of value "GUID1." The first user's photo images are stored in a remote network location, as are all instances of the photo-album application. A second user requesting the photo-album application receives another uniquely identifiable instance of the application and can not view the first user's photos, unless the second user provides the value of the GUID field corresponding to the first user's instance of the photo-album application. The present invention contemplates several methods by which the GUID field is provided to a user.

In one example, the first user transfers the photo-album application to another user, e.g., a third user. That is, the first user composes an e-mail message including a link and droplet to the uniquely identifiable instance of the photo-album application, which has a GUID field of value "GUID1." The link is downloadable and includes the GUID field of value "GUID1" for selectively invoking the unique instance of the photo-album application. In this way, the first user grants access to its personal photos to the third user while substantially prohibiting others (e.g., the second user) from viewing them.

In another embodiment, the first user may view the GUID field and pass it directly to the second, third or both users.

22

When requesting a keyed droplet-enabled application, the system 10 queries the user (e.g., the second or third user) as to whether they want a new instance or a secured instance of the application. If a secured instance is requested, the system 10 prompts the user to enter the identification field value corresponding to the secured application (e.g., GUID1).

Tracking Droplet Dispersion Information:

As noted above, each instance of a droplet is associated to a global unique identification field (GUID). As illustrated in FIG. 8, a data store, shown generally at 640, is maintained at the application server 40 for storing tracking information 642 relating to usage of droplet-enabled applications and/or information. The data store 640 includes, for example, a database, file, link list, or similar storage feature, that contains the GUID and information regarding each client computer 20 that has received links and droplets for droplet-enabled applications and/or information.

By accessing and retrieving the tracking information 642 within the data store 640, droplet dispersion data can be provided upon request. The droplet dispersion data identifies paths that links and droplets of droplet-enabled applications and/or information have taken throughout the system 600. Therefore, the data identifies the links and droplets that are transferred from one user, or client computer, to another (e.g., transmitted via an email message) as well as that are directly accessed and downloaded to client computers from, for example, web sites or imbedded within stand-alone programs.

The tracking information 642 can be used for many purposes, for example, to track the relative success of a promotional campaign. Additionally, the tracking information 642 can be used to report the number of users a droplet-enabled application has reached from a specific, original user.

The tracking information 642 also includes information detailing usage aspects of particular droplet-enabled applications. For example, the tracking information 642 includes information as to when (e.g., time and date) links and droplets corresponding to a droplet-enabled application were downloaded to a client computer, how often (number of times) an application was invoked and for how long each time, what application functionality was utilized during a session, how long an application remained active and visible on a client computer's desktop, how often the application was obscured by another application or minimized, and when the links and/or droplets of the droplet-enabled application were deleted or erased from the client computer.

Generally speaking, a merchant or sponsoring entity of a droplet-enabled application or website providing a droplet-enabled marketing application (survey) may utilize this tracking information to determine, for example, usage patterns of users or to target promotions to users having a link and a droplet corresponding to the droplet-enabled application (such as a droplet-enabled shopping application having a particular item of inventory) resident on its computer for a predetermined period of time. The targeted promotion may, for example, offer a discounted price to move the client toward a decision to purchase the item of inventory.

Preferably, the tracking information provided to merchants or sponsoring entities is anonymous (e.g., does not specifically identify a particular user by name or machine address). Moreover, the tracking information may be provided in a real-time manner by utilizing the communication protocol and message format discussed above.

Alternatively, the tracking information may be utilized by droplet application developers to optimize performance of the

US 7,502,838 B2

23

droplet and/or to evaluate user interest as to particular functionality with the droplet-enabled application.

Providing a Ubiquitous Event Channel:

In one aspect of the present invention, illustrated in FIG. 6, the client/server system 10 includes an event channel 400 for transmitting information to all clients of the system 10, all clients that have downloaded a link and droplet of a particular droplet-enabled application or a specific client computer, for example, the client computer 20. The information includes, for example, sounds, images, text or any combination thereof, transmitted to users to notify them of an occurrence of an event. The notification to the users may be in real-time (upon the occurrence of the event) or at a predetermined, regularly scheduled time period.

In one embodiment, for example, a droplet-enabled e-mail application is provided. In this embodiment, messages are transmitted over the event channel 400 to notify a user of the receipt of an e-mail message. The event channel 400 permits the receipt of messages whether the e-mail application is actively executing or is not. Preferably, and when the e-mail application is not actively executing, the event channel notifies the user of the receipt of a message by flashing or blinking an icon associated to the e-mail application (e.g., icon 320 for the Stock Watcher application 190) on the desktop of the user's client machine. In this respect, the droplet presentation client includes an alarm application. The alarm application periodically evaluates the event receiving buffer 420 at the client computer 20 for messages such as, for example, an "e-mail received" notification message as discussed above. In response to a received message, the alarm application alerts a user of the client computer by, for example, utilizing features of the user interface to blink an icon, activate a sound, or launch a pop-up message or menu. Alternatively, in a MICROSOFT® Windows™ environment, an envelope symbol or icon representing that the message was received is presented in the "tool tray" portion 340 of the desktop (FIG. 5).

The implementation of the event channel 400 requires the cooperation of the client computers, a central event server and application servers spread over the network 50. In FIG. 6, the central event server is implemented within the content provider 30, although it should be appreciated that other implementation strategies are contemplated. The content provider 30 includes an event transmitting buffer 410 operatively coupled to each application server, for example the application server 40, and event receiving buffers within each client computer, for example an event receiving buffer 420 within the client computer 20.

In one embodiment, platform specific software for the client computers, the central event server and the application servers may be installed locally or accessed through a link to a droplet-enabled version thereof. On the client computers, the software permits a user to individually grant the server applications access to its event receiving buffer 420. The client software also provides a view of event history, that is, permits the client to recall previously transmitted information (sounds, images, text or any combination thereof).

Once an application (e.g., a droplet-enabled application 41 executing on the application server 40) is authorized by one or more users, the application 41 posts events by transmitting a message to the transmitting buffer 410 of the central event server (e.g., message 430 from the application server 40 to the content provider 30). The message 430 includes, for example, the details of the posted event as well as information identifying the user or users that are to receive the event. The

24

message is then delivered by the central event server to specified clients along the event channel 400.

Preferably, a user may invoke an application by selecting the event information originating from the application. Additionally, application specific information may be transmitted whether the application itself is active or not.

The ubiquitous event channel 400 provides a mechanism by which a droplet-enabled application provider, developer or sponsoring entity may selectively notify a user or users of the droplet-enabled application that new or revised content is available. In one embodiment, the provider, developer or sponsoring entity can transmit a notification event to all users or a particular user of a droplet-enabled application based on, for example, a pattern of usage of the application. In another embodiment, users of client computers 20 may poll or otherwise evaluate the event channel at a predetermined time period to determine whether new or revised information pertinent to a droplet-enabled application loaded therein is available.

In an exemplary interactive, droplet-enabled shopping application provided on a website, a user downloads the application and an item of inventory onto their desktop. As noted above, tracking information may be provided to the entity sponsoring the shopping applications. If a purchasing decision is delayed, pricing and availability information pertaining to the item of inventory may change. By utilizing the tracking information and the event channel 400, a notification message containing the revised pricing and/or availability information may be transmitted to the user. Alternatively, the entity sponsoring the shopping application may wish to favorably influence the user to buy the item of inventory and, therefore, may transmit a notification message having a discounted price to the user over the event channel 400. According to the present invention, such a targeted promotional message transmitted via the event channel 400 would be received at the user's client computer whether the shopping application is currently executing or not.

Persistent State Maintenance:

The present invention provides capabilities for re-establishing a previous operating state of a droplet-enabled application. For example, when a user re-establishes a session with a droplet-enabled application, the state of the application is restored to what it was when the user ended the immediately prior session.

In accordance with the present invention, the application server 40 maintains state information 48 regarding current operating states of each droplet-enabled application in the data store 640 on the application server 40 (FIGS. 1 and 8). The state information 48 is kept after the application is terminated. The state information 48 includes a user identification field such that operating state information 48 for a particular user is retrievable from the data store 640. It should be appreciated that the present invention contemplates a sign-on procedure or similar mechanism wherein a user registers prior to beginning a session. Accordingly, state information corresponding to a operating session of the user may be identified by a user id or the like. As registration procedures are generally known, further details of the procedures are not further discussed herein. Rather, it should be appreciated that such a registration procedure is within the scope of the present invention.

When the same user re-invokes a droplet-enabled application, the state information 48 corresponding to each application the user executed during their last session is reviewed and the state information 48 pertaining to the re-invoked application is retrieved from the data store 640. The retrieved state

US 7,502,838 B2

25

information 48 is delivered to the user as the droplet-enabled application is presented to the client computer 20. In one embodiment, the user is given an option of re-loading or not re-loading the state information 48 corresponding to the last session. As the state information 48 is maintained on the application server 40 and identifiable by an association to a user's registration information, the state information 48 is provided regardless of client computer 20 that a current and the prior operating session were run on. Therefore, even if a user re-invokes a droplet-enabled application from another client machine, the previous state information 48 is accessible and may be provided to the user upon request.

For example, and referring again briefly to FIG. 8, a user first logs on to (registers with the system 600) and invokes the Stock Watcher application 100 at a personal computer in their office, represented as the client computer 608. During this session, the user alters the size of the window presenting the Stock Watcher application 100 and adds a number of personal stock offerings to a list of "watched" stocks under the Stock Watcher application 100. As should be appreciated, the state information 48 is stored in the data store 640 and accessible by referring the user's identification field. The user terminates the current execution of the application 100 and logs off the client computer 608. Later, the user is traveling and has an opportunity to check on the status of the watched stocks. The user re-invokes the Stock Watcher application 100 from a portable electronic device such as, for example, a laptop personal computer, represented as the client computer 606. Because the user had previously transferred a link (e.g., link 632) to the Stock Watcher application 100 at the client computer 606 and, the user can establish the communication connection 618 for retrieving the Stock Watcher application 100. In accordance with the present invention, the state information 48 corresponding to the previous execution of the application 100 at client computer 608 is retrieved from the data store 640 and presented to the user at client computer 606. As a result, the resized window presenting the Stock Watcher application 100 and the newly added personal stock offerings are presented to the user. Accordingly, the present invention avoids a perceived disadvantage in conventional state tracking methodologies, for example, the cookie technology outlined in the Background Section of this Specification. As discussed above, the conventional methodologies are limited in that prior state information is generally stored on the client computer 20 that previously invoked the session and, therefore, modifications made to an application invoked by a user at one client computer are not apparent when the user the user re-invokes the same application at another client computer.

Preferably, the operating session state information 48 includes sizes and locations of open windows and controls, the position of the cursor within the application, any completed or partially completed data field values, and data fields or items of, for example, inventory selected by the user during the previous session. Preferably, the application server 40 stores the state information 48 in a long-term storage device to ensure that a prolonged delay in user activity may be accounted for. Additionally, a standard API is defined for creating a storable packet of information representing the complete state of a droplet-enabled application. Also, the software running on the client computers 602-608 must be capable of responding to requests (made under the communication protocol outlined above) to create windows and to move the selection.

If an operating session between a user and a droplet-enabled application is active and the user attempts to establish another session with the same droplet-enabled application,

26

the system, 600 identifies a conflict. That is, a message is transmitted and displayed on the client computers 602-608 providing the user the option of either terminating the attempted new session and resuming the existing session, or terminating the existing session and initiating the new session.

In order to identify concurrent operating sessions the system 600 (and system 10) stores a flag indicating that a particular user already has an open session with a particular application. When a user attempts to connect to a new application, the system 600 checks for the value of the flag. If it is set, the system 600 offers the choice to terminate the existing session or create a new one. If the user chooses to terminate, the system notifies the running application that it has been terminated. If the user chooses to start a new session, the system 600 stores a flag indicating the choice. Preferably, the flag includes a timestamp to uniquely identify sessions. If the user attempts a connection and there are multiple active sessions, the user is given the option to terminate any one of them.

Mirroring the Desktop Application on to a Webtop:

In accordance with the present invention, a webtop is a password-protected personal repository, e.g., an account on a web site, for storing links to droplet-enabled applications and information commonly used by authorized users of the account. The links include addresses of servers and names of the processes executed thereon in support of the applications and information in the account.

As a link to a droplet-enabled application and/or information is copied to a client computer 20, e.g., added to the desktop, within a window, or to a menu exhibited on the desktop, then the link may also be copied or "mirrored" to the user's webtop. Preferably, the mirroring behavior is under the user's control. That is, a user may specify that the mirroring should happen all the time, never happen or that a message should ask the user whether they want to mirror the link to the user's webtop. If an option is selected to mirror links to the webtop without any interaction with the user, the information necessary to access the webtop account, i.e., an authorized user's name and password, is encoded within the mirroring process.

In a preferred embodiment, the mirroring process is performed regardless of the method used to store the link to the droplet-enabled application or information on the desktop. That is, whether a link is downloaded by dragging a droplet handle out of a web page or by choosing a menu command to add it, mirroring may be invoked. In one embodiment, the user may choose to record a link in their webtop without adding the link to the desktop. Also, the mirroring process distinguishes between instances of the same application served by different servers.

As a user visits their webtop account, the applications stored for that user are presented in groups. Preferably, applications added are automatically put in a "New Applications" group. Users have the option to create new groups and move applications according to their preferences. In one embodiment, there is a web interface on the webtop that enables a client to register that an application should be added to a particular user's repository.

Multiple Platform Implementation of Droplets:

As discussed above, droplet-enabled client computers 20 include the droplet presentation client 25. In accordance with the present invention, the presentation client 25 is a generic, platform independent application program that processes user interface specifications received from the application server 40 and routes user driven events back to the application

US 7,502,838 B2

27

server 40 utilizing the message formats (e.g., event notification and session commands) discussed above.

As the droplet presentation client 25 is platform independent, one instance of the droplet presentation client 25 can execute all instances of droplet-enabled applications whether downloaded from a web page or a standalone application running on the desktop. However, the implementation of the droplet presentational client 25 may differ over various hardware and software platforms. For example, the droplet presentation client 25 is designed to cooperate with many web browsers of differing vendors, operates on non-personal computer platforms such as PalmPilot, internet-enabled radio telephones, Netpliance™, J-Openers™, tablet computers and others. The presentation client 25 also is designed to operate on multiple computer platforms such as personal computers running Windows™ or Linux operating systems, Apple Macintosh and Sun Solaris work stations. Additionally, the server computers 30 and 40 operate on multiple hardware and software platforms such as, for example, Windows NT and Unix server platforms.

In a web based implementation (within a web page), the droplet presentational client 25 is preferably implemented as a Java applet, an ActiveX component, or as a browser plug-in. Each of these implementation technologies presents an interface that is embedded in a web page. The parameters of the HTML tag used to include the droplet presentational client 25 provides the server to connect to and what application to request to the presentational client 25.

In a desktop based implementation, the droplet presentational client 25 is implemented as a standard native executable. The presentational client 25 is launched as the droplet link is invoked and the presentational client 25 receives information from the link, i.e., from a file associated to the link, that tells the presentational client 25 which server to connect to and what application to request.

Security Features:

The droplet-enabled client/server system 10, configured and operating in accordance with the present invention, includes security capabilities for password and encryption-protected storage and transmission of information, that is, information transmitted within the communication protocol, message format and event channel, discussed above, between the client computers 20 and server computers, e.g., 30 and 40.

To substantially protect inappropriate use of passwords, passwords are digested prior to transmission by using MD5 or similar conventions. Similarly, within applications that request passwords, hashing is performed as passwords are not transmitted in plain text, even on encrypted channels. When a password must be typed into a system, the characters are cloaked, i.e., each character display is represented by an asterisk.

Encryption routines are used within the system 10. As part of an initial protocol negotiation between client and server, the server may specify that some or all of the transmissions may be compressed. Where processing power constraints are not prohibitive, the system 10 uses SSL as a standard mechanism to establish common keys. Where processing power constraints are issued, alternative mechanisms and encryption algorithms may be substituted such as, for example, HTTP tunneling. Substitute algorithms are subject to security review and must be as reliable as the primary algorithm.

It should be appreciated that transmitted messaging within a particular droplet-enabled application may be selectively secured or unsecured, at the option of, for example, the droplet application developer or user of the application. Also, the level of security, for example, 1-bit, 30-bit or 128-bit encryption,

28

is selectable at run-time. That is, sensitive information may be transmitted over secured channels, while less sensitive information is transmitted over less secured channels. As is known to those in the art, there are direct relationships between the levels of security and the constraints imposed on the network configuration and desired network performance.

Developing Droplet-Enabled Applications:

Those skilled in the art should appreciate that developing a droplet-enabled application is similar to developing a GUI or desktop application using conventional component based architectures. Well-designed desktop applications isolate routines for processing user interface, business logic and data storage functions into modules. The separation of user interface, business logic and data storage functions into modules affords a greater portability across differing platforms. For example, modifications can be made to the user interface module to account for hardware specific requirements without affecting the business logic and/or data storage modules. Droplet-enabled applications, constructed in accordance with the present invention, include this layered or module approach to application development.

FIG. 7 illustrates logical layers of a conventional application 500 as compared to that of a droplet-enabled application 550. As is shown, the conventional application 500 includes a user interface layer 510, a business logic layer 520 and a database or data storage layer 530. The droplet-enabled application 550 includes a user interface layer 560, a business logic layer 570 and a database layer 580 that execute on an application server and provide functionality to an invoking client computer.

The business logic layers 520 and 570 and the data storage layers 530 and 580 of the conventional application 500 and droplet-enabled application 550 are substantially similar. A difference in the layered approaches of the applications 500 and 550 is that the user interface layer 560 of the droplet-enabled application 550 includes two components, a client user interface 562 and a server user interface 564. The client user interface 562 executes on the client computer 20, while the server user interface 564 executes, with the business logic layer 570 and data storage layer 580, on the application server 40. Application drivers within the server logical layers (564, 570 and 580) utilize the network communication protocol (i.e., event notification messages, update and window command messages) such that droplet-enabled applications executing on the application server 40 appear to the user to be executing at the user's client computer 20. For example, the application drivers provide instructions for rendering graphical objects, default parameters or data values displayed within the droplet-enabled application 550 to the client user interface 562.

It should also be appreciated that an application that includes the above-described module or layered approach may be rewritten as a droplet-enabled application without substantial rewriting of existing codes. That is, the existing application user interface layer need only be modified.

Preferably, droplet-enabled applications are deployed as dynamically loadable libraries that, once invoked locally by a client computer, are loaded and run under the control of application drivers on a remote server such as, for example, the application server 40. The droplet-enabled applications and the application drivers are preferably written in the Java, C++, Visual Basic, or other equivalent programming languages that support COM or CORBA interfaces. Also, the droplet-enabled application is implemented as an object that is instantiated at startup, holds all program information while active, and causes the application to terminate when the

US 7,502,838 B2

29

object is destroyed. By instantiating many such objects within a single application server, the server can serve any number of users running independent instances of the same application from the same server.

Conclusion:

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications, as will be apparent to those of skill in the art, may be made without departing from the spirit and scope of the invention. By example, and as discussed above, the teachings of this invention are not intended to be limited to any specific method of providing interactive links to applications and/or information stored at remote locations over the network. That is, it should be appreciated that the features disclosed herein are not limited to application in an Internet or web based system.

For example, it should be appreciated that the client computer 20 may be any type of computing device such as a personal computer, work station, portable or handheld device, or the like. In a preferred embodiment, the client computer 20, the content provider 30 and the application server 40 are operatively coupled over a packet-switched network such as, for example the Internet, having wired and wireless interconnections between devices.

Additionally, the preferred embodiment of the present invention includes two server computers, the content provider for providing informational content and the application server for providing droplet-enabled applications and information including rendering instructions, default data values and application-specific business logic to the droplet-enabled applications. It should be appreciated that it is within the scope of the present invention for one server computer to provide these services and functionality. That is, the content provider may include hardware and software resources for providing the droplet-enabled applications and information including the rendering instructions, default data values and application-specific business logic to the droplet-enabled applications. In this embodiment, the two network connections discussed above as communication connections 52 and 54 may physically be one connection that satisfies two logical functions as illustrated herein.

Accordingly, the invention as set forth in the appended claims is not limited to the precise details of construction set forth above as such other variations and modifications as would be apparent to one skilled in the art are intended to be included within the spirit and scope of the invention as set forth in the defined claims.

What is claimed is:

1. A method for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the method comprising:

retrieving, in response to a request of a client computer, a content item having computer program code embedded therein; execution of the embedded computer program code establishing a communication connection to a host computer;

sending operating environment information regarding the client computer from the client computer to the host computer;

retrieving presentation information to present an application and content, the presentation information being based on the operating environment information and comprising at least one of instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the presented application; and

30

presenting, at the client computer, the application and the content based upon the presentation information.

2. The method of claim 1 comprising:

storing, on the client computer, a link for re-establishing the communication connection to the host computer; retrieving the presentation information; and presenting the application and the content.

3. The method of claim 2 comprising allowing a user of the client computer to drag and drop the link for storage at the client computer.

4. The method of claim 2 comprising storing information to identify the application and the content upon selection the link.

5. The method of claim 4 wherein the information to identify the application comprises at least one of an icon and a label representing at least one of the presented application and the content.

6. The method of claim 2 wherein storing further comprises storing instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.

7. The method of claim 2 comprising:

storing, at the host computer, information representing a first operating state of the application and the content presented to the user; and

in response to selection of the link, presenting to the user the first operating state information.

8. The method of claim 7 wherein the first operating state information is presented to the user at a given one of the plurality of client computers.

9. The method of claim 2 comprising:

transmitting the link to a second client computer such that the second client computer is operative to establish a communication connection to the host computer; transmitting presentation information from the host computer to the second client computer; and presenting the application and the content on the basis of the presentation information.

10. The method of claim 9 comprising storing information to track transmission of the link between the plurality of client computers.

11. The method of claim 10 wherein the information to track transmission of the link comprises a global unique identifier that is assigned to the link and information for identifying a given client computer that has received the link.

12. The method of claim 1 wherein the networked computer processing system includes an Internet connection and the client computer includes a desktop-based repository for storage of one or more links to one or more applications and content.

13. The method of claim 12 wherein the networked computer processing system includes an Internet-based repository for storage of the one or more links to one or more applications and content.

14. The method of claim 1 wherein the networked computer processing system includes a desktop-based repository for storage of one or more links to the applications and the content and an Internet-based repository for storage of the one or more links, and wherein the link is stored in both the desktop-based repository and the Internet-based repository.

15. Computer readable media for the storage of program code that when executed by a programmable processor instructs the processor to execute a method for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the computer readable media comprising:

US 7,502,838 B2

31

program code for retrieving, in response to a request of a client computer, a content item having computer program code embedded therein, the execution of the embedded computer program code establishing a communication connection to a host computer;

program code for sending operating environment information regarding the client computer from the client computer to the host computer;

program code for retrieving presentation information to present an application and content, the presentation information being based on the operating environment information and comprising at least one of instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the presented application; and

program code for presenting, at the client computer, the application and the content based upon the presentation information.

16. The computer readable media of claim 15 comprising: program code for storing, on the client computer, a link for re-establishing the communication connection to the host computer;

program code for retrieving the presentation information; and

program code for presenting the application and the content.

17. The computer readable media of claim 16 comprising program code for allowing a user of the client computer to drag and drop the link for storage at the client computer.

18. The computer readable media of claim 16 comprising program code for storing information to identify the application and the content upon selection the link.

19. The computer readable media of claim 18 comprising program code for storing at least one of an icon and a label representing at least one of the presented application and the content.

20. The computer readable media of claim 16 wherein the program code for storing further comprises program code for storing instructions for rendering components of the application, default parameters and data values exhibited within the components, and application-specific business logic for processing input to the application.

21. The computer readable media of claim 16 comprising: program code for storing, at the host computer, information representing a first operating state of the application and the content presented to the user; and

program code for, in response to selection of the link, presenting to the user the first operating state information.

22. The computer readable media of claim 21 comprising program code for presenting the first operating state information to the user at a given one of the plurality of client computers.

23. The computer readable media of claim 15 comprising: program code for transmitting the link to a second client computer such that the second client computer is operative to establish a communication connection to the host computer;

program code for transmitting presentation information from the host computer to the second client computer; and

program code for presenting the application and the content on the basis of the presentation information.

32

24. The computer readable media of claim 23 comprising program code for storing information to track transmission of the link between the plurality of client computers.

25. The computer readable media of claim 24 comprising program code for storing a global unique identifier that is assigned to the link and information for identifying a given client computer that has received the link.

26. The computer readable media of claim 15 comprising program code for maintaining an Internet connection and a desktop-based repository for storage of one or more links to one or more applications and content at the client computer.

27. The computer readable media of claim 26 comprising program code for maintaining an Internet-based repository for storage of the one or more links to one or more applications and content.

28. The computer readable media of claim 15 comprising program code for maintaining a desktop-based repository for storage of one or more links to the applications and the content and an Internet-based repository for storage of the one or more links, and wherein the link is stored in both the desktop-based repository and the Internet-based repository.

29. A system for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the system comprising:

a presentation client executing at a client device operative to transmit operating system environment information to the application server, receive presentation instructions from the application server to display one or more content items, transmit one or more events to the application server, receive updated presentation instructions from the application server to display the application and one or more updated content items in accordance with the updated presentation information and maintain an operating state of the application; and

the application server operative to transmit presentation information to the presentation client for display of the application, the one or more content items and the one or more updated content items on the basis of the operating system environment information, execute application specific business logic on the basis of the one or more events to generate the updated presentation instructions for transmission to the client player.

30. The system of claim 29 wherein the application server is operative to maintain the operating state of the presentation client.

31. The system of claim 29 wherein the application server is operative to maintain one or more default data values.

32. The system of claim 31 wherein a given default data value is a parameter to the application specific business logic.

33. A method for presenting an application in a networked computer processing system having a plurality of client computers and a plurality of host computers, the method comprising:

initiating a connection from a presentation client to an application server, the presentation client operative to transmit client operating system environment information to the application server;

transmitting presentation instructions and one or more content items from the application server to the presentation client, the presentation instructions based on the operating system environment information and operative to instruct the presentation client as to the display of the application and the one or more content items;

transmitting one or more events from the presentation client to the application server; and

US 7,502,838 B2

33

on the basis on the basis of the one or more events, executing application specific business logic at the application server to transmit updated presentation instructions and one or more updated content items to the presentation client for display with the application.

34. The method of claim 33 wherein initiating comprises transmitting information regarding the presentation client to the application server.

35. The method of claim 33 wherein transmitting a given event comprises transmitting a user event.

34

36. The method of claim 33 comprising storing operating state information regarding the presentation client.

37. The method of claim 36 comprising storing the operating state information at a device upon which the presentation client is executing.

38. The method of claim 36 comprising storing the operating state information at the application server.

* * * * *

**CERTIFICATE OF COMPLIANCE
WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS,
AND TYPE STYLE REQUIREMENTS**

1. This brief complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B) because it contains 6,452 words, as determined by the word-count function of Microsoft Word 2010, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii) and Fed. Cir. R. 32(b).

2. This brief complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the type style requirements of Fed. R. App. P. 32(a)(6) because it has been prepared in a proportionally spaced typeface using Microsoft Word 2010 in 14-point Times New Roman font.

/s/ Daniel L. Geyser

Daniel L. Geyser

Principal Attorney

STRIS & MAHER LLP

725 S. Figueroa Street, Suite 1830

Los Angeles, CA 90017

Tel.: (213) 995-6811

Fax: (213) 261-0299

daniel.geyser@strismaher.com

Counsel for Appellant Droplets, Inc.

December 9, 2016

CERTIFICATE OF SERVICE

I hereby certify that on December 9, 2016, an electronic copy of the foregoing Opening Brief was filed with the Clerk of Court for the U.S. Court of Appeals for the Federal Circuit, using the appellate CM/ECF system. I further certify that all parties in the case are represented by lead counsel who are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

/s/ Daniel L. Geyser
Daniel L. Geyser
STRIS & MAHER LLP
725 S. Figueroa Street, Suite 1830
Los Angeles, CA 90017
Tel.: (213) 995-6811
Fax: (213) 261-0299
daniel.geyser@strismaher.com

December 9, 2016